A WISWESSER FORMULA DECODER AND GRAPHIC DISPLAY

Willy Verbeetel
B.Sc., Sir George Williams University, 1972

A Thesis

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements for
the Degree of Master of Computer Science at
Concordia University
Montreal, Canada

February, 1977

ABSTRACT

WILLY VERBESTEL

A WISWESSER FORMULA DECODER AND GRAPHIC DISPLAY

This thesis develops a new computer program to decode a Wiswesser line notation and to display on an interactive graphic terminal a structural notation of the chemical code.

The program, coded in FORTRAN IV, may be considered as several independant modules, input of a Wiswesser code, the ring and the chain decoder, conversion of connecting table into a drawing table, and finally the display of the structural notation on a TEKTRONIX graphic terminal.

The load modules created on a XEROX Sigma9 occupies 9K words and the CPU time required varies between 0.3 and 1.2 seconds.

In order to keep the core size as well as the CPU time within reasonable limits the program is not able to handle multipliers and multiple bridges over ring structure.

ACKNOWLEDGEMENTS

I should like to express my appretiation to Professor Dr. C.Y. Suen who served as thesis director and to Mr. Glen Lockwood who offered me the use of the Xerox computer facilities of the Communication Reseach Centre to prepare this thesis.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

1.1 INTRODUCTION.

A chemical compound may be identified in many ways, the most familiar of which are the chemical nomenclature, the structural notation, and the linear expression.

The first chemists gave trivial names to the few known compounds. These names were translated into many languages and serve as a base for the modern chemical nomenclature, which uses names to define a chemical compound. This is the oldest and the most widely used method for publications, patent registrations, and commercial catalogues.

Unfortunately the modern chemical nomenclature shows a great redundancy and for this reason is unsuitable for unique identification of chemicals. Computer based information systems must use a more precise method of chemical identification.

The key idea (1) of structural notation in chemistry was popularized in 1860, when the leading chemists of the world attended the first International Chemical Congress at Karlsruhe to resolve their confusions about atoms, molecules, and equivalents.

The exact convention relating to "Linear expression of formulae" was clearly defined by A. D. Mitchell (2). "Each full point (or period mark) is regarded as separating two atoms which are directly linked in the main chain of a compound, and atoms or groups attached to each of these atoms are written immediately after it and before the next full point." Then in 1863 Emil Erlenmayer (3) omitted the punctuation resulting in the present common form. And in 1866 August Kekule (4) proposed the cyclic representation of chemicals and the electron delocalisation as used now.

Although virtually all the main ideas relating to line-formula descriptions were conceived and published within the short period of seven years after the birth of structural chemistry in 1861, no basically new practices appeared for some seventy-nine years. Then, within another brief period of seven years (1947-1954), virtually all of the fundamental features of structure-delineating chemical notations appeared in the international chemical literature.

The linear representation : Gordon, Kendall and Davison published in 1948 the G-K-D ciphering method (5), a universal code as an aid to chemical systematics. They designed their system for early versions of computers, and Davison shortly thereafter reported on "Sorting of Chemical Groups Using Gordon - Kendall - Davison Ciphers"(6).

In 1949 the IUPAC (International Union of Pure Applied Chemistry) Commission on Codification, Ciphering and Punched Card Techniques announced its interest in seeking an internationally suitable chemical notation and invited designers to submit their proposals for review in 1951. It enumerated eleven desirable characteristics or "desiderata for an internationally acceptable chemical notation"(7). These were: simplicity of use, ease of printing and typewriting, conciseness, recognizability, ability to generate a unique chemical nomenclature, compatibility with the accepted practices of inorganic chemical nomenclature, uniqueness, generation of an unambiguous and useful enumeration pattern, ease of manipulation by machine methods, and ability to deal with partial indeterminates.

John A. Silk responded to the IUPAC commission's invitation with his "New System of Organic Notation" (not published). J. G. Cockburn responded at the same time (1951) with his "Newcastel System". These linear notations were briefly reviewed by the IUPAC Commission at the Massachusett's Institute of Technology, along with a notation that was based on the "Principle of Least Effort"(8). This notation, first described in 1950, was published in a small manual (9) by W.J. Wiswesser. E. G. Smith, using this notation, wrote a faculty report on its

use in "A Punched Card Catalog of the Physical Properties of Some Common Organic Chemicals" and after seven years of hard labour, wrote a book published by McGraw Hill Book Company (10). This is available with additional Computer-Generated Services from the Institute for Scientific Information.(11)

1.2 REQUIREMENT for a DISPLAY.

Every representation of a chemical compound is an abstraction whose value is a function of its usefulness for a particular application. The most widely used chemical representation is the structural diagram, which might well be termed the natural language of chemists. Many representations are useful for special purposes and among these is the Wiswesser Line Notation (WLN), which has been found to have advantages in information retrieval applications because of its compact linear form.

Information chemists are a certain class of chemists whose principal occupation is searching the chemical literature manually or using a computer data-bank. Only a small group of 'information chemists' is willing to deal with a linear representation of molecules because this form does not represent the natural language of chemists.

Furthermore any type of computer research system giving information directly to the bench chemist should display the end result in structural form.

1.3 HISTORY of the DISPLAY.

In 1968 Hyde and Thomson reported the necessity of such a structural display as the end point of searching and proposed his "Structure Display" (12), a COBOL program to plot a structure from a connectivity matrix derived from the Wiswesser notation. This computer program limits the representation of Wiswesser notation to a simple ring with side chain and contains many restrictions about the orientation of rings and substituents on the ring. The production of the connectivity matrix by computer from a Wiswesser notation was earlier reported by Hyde, Wiswesser et al(13).

In 1969 Ronald Gottardi(14) proposed a new set of graphics characters called "octobliques" for input and output of structural diagram on a standard impact-type, high-speed computer printer. His interest was limited to the reproduction of graphic structures and no subsequent papers were written on his system.

In 1970 Tarell et al proposed a method of generation of Wiswesser Line Notation(15), using a FORTRAN program written for a PDP-10 computer. He believed that most chemists would prefer to interact with the computer in their natural language and provided a program to produce a (WLN) Linear Notation from a structure diagram drawn on a Rand tablet.

M. A. T. Rogers(16) presented in 1970 the ICI(Imperial Chemical Industries Ltd) system CROSSBOW(Computerised Retrieval of Organic Structures Based On Wiswesser) which included as an output requirement a modified print chain to print a stylised skeleton from a CAS(Chemical Abstract Service) connexion table.

Feldman et al presents in a series of papers a substantial work in searching chemical files and ending these computer searches with a production of diagrams. The first paper about structural display of Wiswesser code appears in 1971 with his "Interactive Searching of Chemical Files and Structural Diagram Generation from Wiswesser Line Notation"(17). This system of programs contains a FORTRAN program to generate a two-dimensional representation of chemical structure directly from the WLN on a normal terminal using different symbols to simulate chemical bonds. He recognised this to be a poor chemical representation

compared to the CROSSBOW system described earlier. The second paper about the subject appears in the same year with his "An Application of Interactive Computing - A Chemical Information System"(18) where he starts to display structures on a screen but does not work on the presentation of the molecule. One month later he presents(19) a new type of searching technique based on CAS connexion tables (described earlier). This last system does not improve the quality of graphic output and his paper does not define the type of screen used in the representation.

George A. Miller in 1971 presents a program for encoding and decoding(20) WLN using a Rank tablet as inout and an undefined screen as output device. He reports a series of restrictions such as limitation to unbranched perifused rings.

Alfred Feldman reports in 1973(21) on "A Chemical Teletype" a Teletype model 37 modified for the coding of chemical structures.

And finally a very complete paper on decoding and graphic representation of chemicals was presented by Leo in 1973 in a series of two publications(22,23). He devoted such a large part of the programming to trying to decode a multiplier that he had to restrict himself to only the simplest expressions, and he found this unrealistic to use.

A complete literature search in the field does not show any more work published from that period to the present and the lack of any general Wiswesser decoder and display system suggested to me the subject of this thesis.

CHAPTER 2.

PHILOSOPHY of the PROGRAM

A FORTRAN program has been written to decode a Wiswesser line notation and display on an interactive graphic terminal a structural notation of the chemical code entered.

The program was implemented on a XEROX Sigma9 computer and the terminal used was a Tektronix 4015 using TCS (Tektronix Control System software) as a graphics library.

The program may be considered as five modules:

1. Input and mapping module.
2. Ring decoding module.
3. Chain decoding module.
4. Conversion from connecting table to drawing table.
5. Module to display on the graphic terminal the structural notation.

Each of these modules will be documented briefly and some of the algorithms used will be described. A more detailed description may be found in Appendix 1.

2.1 Input and mapping module.

The program is coded to allow a manual entry of a Wiswesser code. It may be modified to receive codes from any kind of host retrieval system.

The input string of characters is mapped in a string of integer numbers in order to facilitate the decoding process, to detect the presence of an illegal Wiswesser character (see valid list under rule 2 in chapter 3), and finally to reduce conversion difficulties between computers of different manufacturers.

The mapping is made as follows:

a. Numerics are converted to their integer values (e.g. 0-9).

b. Space, ampersand, dash and slash are converted to integers 10-13 respectively.

c. Alphabetics are mapped into the range of values 14

to 39.

2.2 Ring decoding module.

a. Detect the boundary of ring code definitions as explained by rule 31 in chapter 3.

b. Construct a ring table containing for each ring found, the earliest locant, the ring size and a saturation flag.

c. Construct a special link stack showing the start and end locants-position for the special link to be created.

d. Construct a bridge stack containing the locants included in a bridge over a ring.

e. Construct a multicyclic stack containing the locants of multicyclic atoms in the molecule.

f. Construct a perifused stack containing the locants of perifused atoms.

g. From the tables and stacks previously created, a connecting table has been built, where an entry is provided for each element considered as a node in the display. (e.g. a carbon in a ring, a (CH2)3 for a chain).

Information is provided as follows for each column.

Position	Description
1	Pointer for the next free entry to be used for the node joining this entry.
2 - 3	X and Y coordinates for this node (filled during execution of the 4th module).
4 - 5	Pointer for the position in the table of the next node, and the absolute angle in degrees to be used to compute the absolute position of the next atom from the position of the current one.
6 - 11	Three two-word entries with the same definitions as 4 and 5 for extra bonds. (e.g. for each entry in the connecting table the program checks first the contents of cells 4 and 5 and if they are not zero, a right shift before filling cells 4 and 5 is performed).
12	Contains a flag counter for saturation set at zero for an Sp3 element, at 1 for an Sp2 and at 2 for an Sp. (Sp are chemical units of unsaturation).
13	Contains the number of bonds already used in the outside of the ring to allow computation of the

outside direction.

14 Direction (between 0 and 360 degrees) for the bond to the outside of the ring. The algorithm used is: The first time the node is used the direction is calculated with the equation Θ + (360/N) + 180 where Θ is the angle to the next atom and N is the size of the ring. For repeated uses of a node the direction is the complement of the angle used to join the two rings.

15 The number of characters to be plotted at the node, excluding the C and H which are not represented in ring notation. The characters to be plotted are stacked in a vector.

16 Position of the first character to be plotted in the stack.

17 Flag to be used for multicyclic points.

18 Flag for multiplication of the atom.

19 Flag for a multiple entry in the chain structure, used to compute the direction to be used.

20 Flag for locating a spiro atom.

When information in the connecting table is provided for the ring nodes, the ring code is scanned for heteroelements which are stored in the connecting table as they are found.

2.3 Chain decoding module.

This module is used for the decoding process of a chain of elements or chains of substituents on a previously defined ring notation. A stack of pointers is used to keep track of branching segments.

A complex algorithm had to be written to display numerals in (WLN) notation or carbohydrate segments.

(WLN) notation	At beginning of the code	At middle of the code	At the end of the code
1	CH3-	-CH2-	-CH3
2	CH3CH2-	-(CH2)2-	-CH2CH3
3	CH3(CH2)2-	-(CH2)3-	-(CH2)2CH3
1U or 1UU	CH2= or CH▪	-CH= or -C▪	invalid
U1 or UU1	invalid	=CH- or ▪C-	=CH2 or ▪CH
U1U or UU1U or U1UU or UU1UU	invalid	=C= rest invalid	invalid
2U or 2UU	CH3CH= or CH3C▪	-CH2CH= or -CH2C▪	invalid
U2 or UU2	invalid	=CHCH2- or ▪CCH2-	=CHCH3 or ▪CCH3
U2U or UU2U or U2UU or UU2UU	invalid	=CHCH= or ▪CCH= or =CHC▪ or ▪CC▪	invalid
3U or 3UU	CH3CH2CH= or CH3CH2C▪	-(CH2)2CH= or -(CH2)2C▪	invalid
U3 or UU3	invalid	=CH(CH2)2- or ▪C(CH2)2-	=CHCH2CH3 or ▪CCH2CH3
U3U or UU3U or U3UU or UU3UU	invalid	=CHCH2CH= or ▪CCH2CH= or =CHCH2C▪ or ▪CCH2C▪	invalid

2.4 Conversion of the table.

The first step will provide in columns 2 and 3 of the connection table the absolute coordinates on the screen for the atom to be plotted.

This is computed assuming an arbitrary position of (500,365) for the first atom of the connecting table and the other position is determined by the sine or cosine of the angle stored in the table and a bond length of 60 raster units (on CRT) for a ring bond and 50 raster units for a chain bond. These bond sizes are set to differentiate an overlap of a chain on a ring.

The second step will provide a drawing table using the position of atoms to be linked.

This table will have a row entry for each line segment to be plotted and will contain in columns 1 and 2 the absolute coordinates of the beginning of the segment, and in columns 3 and 4 the absolute coordinates of the end of the segment. The unsaturation will provide a smaller bond in the drawing table. Computation is done to start or stop the drawing beam at the edge of an imaginary box around any string of characters.

Finally using the drawing table the structure is displayed on the graphic terminal.

2.5 The drawing module.

This module will scan the drawing table to find maximum and minimum of each pair of coordinates and frame the virtual window to provide a pleasant display on the screen.

Then it displays all segments as extracted from the drawing table and finally scans the connecting table to locate the string of characters to be plotted.

CHAPTER 3.

RULES AND DESCRIPTION OF ALGORITHM.

The decoding of WLN format follows a fixed set of rules, these rules must be compatible with encoding rules used to produce the WLN.

In this chapter we employ a set of encoding rules to generate a program for decoding.

The following rules were extracted from the book published by E., G. Smith(10), they govern the actual WLN encoding of chemical compounds. These rules, together with comments and notes about the algorithm used to decode each particular case constitute the body of this chapter. Furthermore to present the closeness between the encoder and the decoder the rules listed will be presented under the same classification as proposed by E. G. Smith(10).

3.1 NOTATION SYMBOLS UNBRANCHED CHAINS

Rule 1. Cite all chains of structural units symbol by symbol as connected.

Rule 2. Resolve all otherwise equal alternatives in symbol sequences by selecting the sequence that would give the notation the LATEST position in an alphanumeric list arranged in the sequence:

&-/0123...789101112(etc.)ABCDEF...UVWXYZ

(Earliest position) (Latest position)

Give the blank space the earliest position of all symbols.

The two previous rules imply a sequential decoding process for a chain and this method was applied to the program. Therefore the program may be compared to a one-pass decoder.

3.2 BRANCH SYMBOLS

Rule 3. Use the letter U to denote a double bond in a chain when the double bond is:

(a) between any two atoms where cis-trans or syn-anti isomerism can arise upon proper substitution;

(b) between a carbon atom and any branching, divalent notation symbol:

(c) between any noncarbon atom and an alkyl numeral;

(d) between carbon atoms in allenes and cumulenes. Use the letters UU to denote a triple bond between carbon atoms(an acetylenic bond).

The algorithm to process the U notation is:

1) Provide in the two cells of the connecting table a 1 for unsaturation.

2) Provide in the character stack a string of characters following the table in chapter 2. (Description of chain module).

Rule 4. Indicate all other unsaturations not included in rule 3 by directly joining the symbols of the atoms involved.

The decode process does not show implicit unsaturation, e.g. HCN is the complete formula for cyanhydric acid.

Rule 5. (a) When the hydrogen symbol H is not implied as part of another symbol, suppress the H symbol by

citing it directly after the symbol to which it is attached. Orient the notation after the H symbol is suppressed.

(b) Cite the H symbol after an alkyl numeral representing an unbranched, saturated hydrocarbon.

The hydrogen symbol when found in a chain decoding will be treated as a chain terminator.

Rule 6. Cite branched structures along that chain of notation symbols in the graphic formula which includes, first, the largest possible number of branch symbols, and after this the largest possible number of notation symbols. Start at the end of this chain required by rule 2 and then follow rules 7 and 8.

Rule 7. After each branch symbol, cite first, in the following order of choice,

(a) the symbol chain with the fewest branch symbols; and after this,

(b) the symbol chain with the fewest notation symbols; and after this,

(c) the symbol chain with the latest alphanumeric position (ignoring ring locants, for which see rules 23 and 39).

Rule 8. (a) When more than one symbol chain follows a branch symbol, show the end of each symbol chain, except the last one, by punctuating it with an ampersand, unless the chain ends with one of normally terminal symbols -E, -F, -G, -H, -I, -J (generic halogen), -O, -W, or -Z, in which case no ampersand punctuation is nescessary.

(b) When an E, F, G, I, or J halogen symbol is NOT terminal, enclose it within hyphens.

(c) Punctuate a symbol chain following a branch symbol of variable valence (such as P) with an extra ampersand, in addition to any required by rule 8a, IF the next symbol in the notation is attached to a branch symbol next preceding the one of variable valence. Add an additional ampersand for each additional branch symbol intervening.

A last in first out stack (branching stack) is used to store the position of any branching atom therefore the

position of the closest branching location can be obtained after any segment terminator.

3.3 SYSTEMATIC CONTRACTIONS

Rule 9. Write the graphic formula and apply the basic orienting rule 2 (latest position rule) and rule 6 (branched notation rule) BEFORE making any contractions. Make No contraction that gives an ambiguous notation. Never omit the methyl symbol when it initiates the notation.

Rule 10. Omit the methyl group "1" symbols after all X, Y, and K symbols that are not within ring signs but retain the punctuating ampersands. Also omit these ampersands if the methyl-branched X, Y, or K symbol is the ONLY branch symbol of any kind in the notation, or if it is the LAST branch symbol in the uncontracted notation for a molecule or ion. See rule 29 for methyl contractions in ring substituents.

At the end of each segment the decoder verifies the application of this rule and if needed a CH3 segment is automatically created.

Rule 11. (a) When a graphic formula contains two or more identical chains of notation symbols attached to another symbol or symbol group, or attached to each other in mirror image order, cite all the repeated symbols only once, followed by a numeral with a blank space before each digit to show how many times the preceding symbol chain is repeated, BUT ONLY IF the resulting notation has at least four fewer marks than the notation before this contraction is made. A mark is any number, punctuation mark, or letter (including ring locants); a space is not a mark.

(b) The multiplier acts on all the preceding notation symbols back to the nearest slash mark (/), if any.

(c) Do not divide and multiply structural units denoted by either an alkyl numeral or the acetylenic UU symbols.

Rule 12. (a) When a notation contains connected symbol groups which are repeated, contract the notation (if this saves four marks or more) by (1) citing the repeated symbol group once, (2) enclosing it

within slash marks, and (3) placing a multiplier after the second slash marks; a multiplier directly following such a slash mark (with no intervening symbols) multiplies the entire quantity within the preceding pair of slash marks.

(b) Use a pair of slash marks to enclose a central asymmetric polyvalent group of notation symbols when such a group connects identical symbol chains that can be multiplied by rule 11a. The symbols between these slash marks must appear in the same order as in the uncontracted notation when it is oriented by rule 2.

(c) When alternative multiplier contractions exist, multiply within slash marks by rule 12a and not centrosymmetrically by rule 11a. Resolve any remaining alternatives by rule 2.

Rule 13. Use successions of multipliers to give cumulative multiplication if each multiplier saves four or more marks. To multiply segments within other multiplied segments, use double slashes (or triple slashes if necessary) around the outermost segments, omitting any slashes that would initiate the notation.

The program does not accept mirror image reduction and requests that the entire code be entered. The WLN notation is not necessarily symmetrical for a symmetrical molecule. The mirror image therefore does not imply that a notation mirror image is to be used in the decoding process. Example of the non mirror code:

WNR CZ F- 2 becomes WNR CZ FR BNW DZ.

3.4 ORGANIC SALTS

Rule 14. Cite two-letter atomic symbols with two capital letters preceded and followed by a hyphen. To allow use of five single-letter international symbols for more efficient purposes, the following new symbols are assigned: -KA- potassium; -UR- uranium; -VA- vanadium; -WO- tungsten; -YT- yttrium.

An algorithm is used to detect this case and the metal is assumed to be bivalent which is almost always true and the code is entered in the character stack.

In the case of a monovalent metal, an end of a chain is indicated, and an error message will be printed for any higher valence.

Rule 15. Denote as separate units the ions of organic salts and the molecules (addends) making up organic addition compounds, each oriented by rule 2 without regard to any ionic links between them. Separate the notations for each different ion or addend with an ampersand preceded by a blank space. If only one ion or addend is organic, cite this first. If more than one of the ions or addends are organic, cite first the one with the most notation symbols, resolving equal alternatives by rule 2. If two or more different ions or addends remain, repeat this procedure, first for organic and after this for all inorganic constituents except water. Cite all OH addends last.

Rule 16. Do not show the postulated proton transfers on organic base salts or salts of unsubstituted ammonia (ZH), but cite these compounds as addition compounds by rule 15.

Rule 17. Denote tertiary sulfur, quaternary nitrogen and analogous "onium" compounds as organic ionic salts with the "space ampersand" punctuation.

The chain decode module creates for this case a special bond labelled as (+-) that is used to show any salt bond.

Rule 18. (a) In compounds containing ions or addends with organic constituents, a multiplier acts ONLY on the ion or addend it FOLLOWS.

(b) When the entire notation of an ion or addend must be multiplied and another ion's or addend's notation follows, place the multiplier of the earlier ion's or addend's notation AFTER the ampersand that separates it from the notation of the later ion or addend.

(c) When an ion's or addend's notation has been multiplied by rule 18b and the following ion's or addend's notation must also be multiplied, separate the first multiplier from the following notation symbol by interposing a slash mark to stop the action of the next multiplier.

Multipliers are not accepted see rules 11-13 above.

Rule 19. In compounds denoted as organic salts or addition compounds by rule 15, denote the following inorganic moieties by the rules for inorganic compounds,

(a) any metallic ion that is bonded to other inorganic atoms or ions;

(b) any inorganic moiety whose structure is not stated;

(c) any entirely inorganic addend except the binary or oxy-acids containing one atom of sulfur, phosphorus, nitrogen, or a halogen.

The decoder uses the same method in this case as for the normal salt explained for rules 15-17.

3.5 BENZENE DERIVATIVES

Rule 20. Subordinate the benzene symbol R to all other notation symbols, including the alkyl numerals, by ranking it in earlier position than the ampersand,&.

Rule 21. Denote disubstituted benzene compounds by the following steps.

(a) Ignoring ring positions, write out the notation, oriented by rule 2, leaving room for later insertion of a locant before the symbol of the substituted group appearing after the R symbol in this oriented notation.

(b) In the structural formula, assign the "a" locant to the ring position of the substituent that precedes the R symbol in the oriented notation. If the oriented notation leaves alternate starting points in the structural formula, resolve them by rule 23.

(c) Assign consecutive letter locants to consecutive ring atoms around the ring in the direction that will give the earliest locant to the substituent that follows the R symbol in the oriented notation from step (a).

(d) In the oriented notation from step (a), insert before the symbol of the substituent that follows the R symbol its ring locant determined in step (c) a capital letter B,C, or D preceded by a blank space. The "a" locant is not cited but is expressed in the notation by the direct joining of the symbol of the substituent at that position to the R symbol following it.

3.6 MULTISUBSTITUED BENZENE RINGS

Rule 22. Denote multisubstituted benzene compounds by the following steps.

(a) Ignoring ring positions, write the graphic formula, arranging the symbol chains of the benzene substituents around the R symbol as a branch symbol.

(b) Ignoring ring positions, cite the notation along that chain of notation symbols in the graphic formula which includes, first, the largest possible number of branch symbols, and after this the largest possible number of notation symbols (rule 6). Start at that end of this chain having latest alphanumeric position (rule 2).

(c) After each branch symbol (including a branched R) cite first, in the following order of choice, the symbol chain that has (1) the fewest branch symbols; and after this (2) the fewest notation symbols; and after this (3) the symbols with latest alphanumeric position. Write out the oriented notation, leaving room for later

insertion of the ring locant before the first symbol of each substituent chain that follows an R symbol.

(d) In the structural formula, assign the "a" locant to the ring position of the substituent that precedes the R symbol in the oriented notation from step (c). If the oriented notation leaves alternate starting points in the structural formula, resolve them by rule 23.

(e) Assign consecutive letter locants to consecutive ring atoms around the benzene ring in the direction that will give the earliest locant to the first substituent that follows the R symbol in the oriented notation from step (c). If this leaves equal alternatives for choosing the direction of the locant path in the structural formula, resolve them by rule 23.

(f) In the oriented notation from step (c), insert the ring locant determined in step (e) before the initial symbol of each substituent chain that follows an R symbol. Omit the "a" locant as required by rule 21d.

(g) Make all contractions AFTER all orientations have been determined (rule 9).

Rule 23. If equal alternatives remain for the choice of the starting point in rule 21b or 22d, or for the direction of the locant path in rule 22e, resolve them by choosing the alternative that will produce the notation with the earliest set of ring locants-i.e., the notation that would appear first in an alphabetically arranged list of notations.

Benzene notations are treated as any other ring notation but processed by the Chain Module because a benzene code is always a 6 membered unsaturated ring without any heteroelements.

3.7 BENZENE RINGS IN BRANCHING CHAINS

Rule 24. Put an ampersand at the end of each symbol chain (except the last one) after each branch symbol if the chain contains a heterocyclic, carbocyclic, or benzene ring, even though the chain ends with one of the terminal symbols of rule 8. Add an additional ampersand for each additional ring in the chain. To determine the length of these chains for orientation

purposes, count the symbols in the graphic formula and do not count either the ring locants or the final ampersand(s).

An ampersand character is used in the same manner as any other end of segment.

3.8 RING CONTRACTIONS

Rule 25. Use numerals preceded by a space as multipliers (as limited by rule 11) to denote the multiplication of a number of large, identical symbol groups substituted on a benzene ring, placing the locants between the ring symbol R, and the multiplier. Omit the first locant, "a," when the multiplied symbols precede an R symbol.

Rule 26. When multiplied groups follow a ring symbol (heterocyclic, carbocyclic, or benzene) and rule 27 does not apply, add a hyphen after each locant to which the multiplied group is attached to distinguish these locants from the methyl contractions of rule 29. To stop the multiplier action, interpose a slash

between the hyphen of the last such locant and the multiplied symbols.

Rule 27. When the multiplied symbols fill all the benzene ring positions not specified for other groups, the locants of the multiplied symbols are omitted and replaced with a hyphen. Interpose a slash mark between the hyphen and any multiplied symbols that follow the hyphen.

Rule 28. When any notation containing ring symbols is multiplied, the multiplier acts on the structural unit being described and not on the locant sets. If the multiplier immediately follows a locant, interpose a hyphen.

Multipliers are not allowed. See rules 11-13.

Rule 29. (a) Omit the methyl group "1" symbol when it follows a ring locant.

(b) Omit the methyl group "1" symbols after a terminal X, Y, or K symbol in a ring substituent chain. Also omit the punctuating ampersands unless the X, Y, or K symbol is the LAST cited

substituent on a ring which is in a side-branch chain.

The decoder will provide a CH3 segment automatically in a way described earlier.

3.9 CYCLIC COMPOUNDS OTHER THAN BENZENE

Rule 30. To denote cyclic compounds other than benzene, assign consecutive letter locants to consecutive ring atoms in such a way as to achieve the longest possible chain of consecutive ring locants that does not cross a fused ring junction, and that makes it possible, in the following order of choice, to

(a) cite the SMALLEST rings present (following rule 31b) to defin

(b) cite the FEWEST rings necessary (following rule 31b) the structure completely and to make any branch locants citable;

(c) have the fewest branch ring locants, all of which must be citable; and after this to attain

(d) the lowest sum for the fusion locants;

(e) the earliest alphabetic set of fusion locants in the order of their appearance in the notation (following rule 31b);

(f) the earliest set of notation symbols for denoting bridges and nonconsecutive locant pairs;

(g) the earliest sequence of ring numerals (as 56 before 65);

(h) the earliest set of locants in the ring for cited ring segment symbols, excluding U;

(i) the earliest alphabetic set of cited ring segment symbols, excluding U;

(j) the earliest set of locants in the ring for cited U symbols;

(k) the earliest locants for H-symbol ring saturation signs (rule 32a);

(l) the earliest sequence of & and T ring saturation signs (rule 32d);

(m) the earliest set of locants in the ring for substituents on the ring;

(n) the earliest set of locants in the notation for substituents on the ring. If equal alternatives still remain, resolve them by assigning the earliest locant to the substituted group symbol(s) with latest alphanumeric rank.

Rule 31. Delineate ring structures other than benzene with notation symbols in the following order.

(a) Begin notations for carbocyclic ring structures with the symbol L; begin notations for heterocyclic ring structures with the symbol T.

(b) Next cite numerals showing the number of ring atoms in each ring, citing these numerals in the order in which each ring is completed by the locant path. Before each ring numeral cite the earliest locant in that ring, but omit any "a" locants.

(c) Cite next, without spaces, any pair(s) of nonconsecutive locants required by rule 43, as follows: (1) In each pair cite the earlier locant first. (2) Cite all the locant pairs in the order in which they are completed by the locant path. (3) Insert a slash mark before each locant pair. (cc) Next cite, in ascending alphabetic order, and each preceded by a blank space, the locants for any bridge atoms, repeating any bridge locant as many times as it is shared by more rings than two.

(d) If any multicyclic point atoms are present, next leave a blank space and cite a numeral showing the number of multicyclic point locants which follow, citing these in ascending alphabetic order, WITHOUT SPACES before them, and repeating any multicyclic point locant as many times as it is shared by more than three rings. If rule 30 leaves equal alternatives, cite the earliest set of multicyclic point locants. Finally, leave a blank space and cite the LAST ring locant in the ring structure.

(e) Next cite, in ascending alphabetic order of their ring locants, any heteratomic, V, and where required, X and Y symbols, with their ring locants where required. Omit an "a" locant ONLY in a MONOCYCLIC notation, joining the ring segment symbol at the "a" position directly to the ring numeral. Cite the other ring segment symbols (excluding U) in connecting order, interposing locants only when the cited ring segments are not directly connected or when branch locants are used.

(f) Cite next any acetylenic UU symbols together with any U symbols required by rule 32, in a

single sequence in ascending alphabetic order of their initial ring locants. After this cite any H,&, and T symbols required by rule 32.

(g) Next add the symbol J to indicate the end of the notation for the ring structure itself.

(h) Following the J symbol, denote the ring substituents, preceded by their ring locants (including "a" locants) and cited IN ASCENDING ALPHABETIC ORDER OF THEIR RING LOCANTS.

(i) If two substituents have the same locant, cite first the substituent having later alphanumeric position.

Rule 32. If any ring in a ring system contains one or more carbon atoms bonded to FOUR other atoms and NOT cited with the X symbol (rule 41a), indicate this as the last item within the ring signs in the following way:

(a) If there is ONLY ONE such carbon atom in any one ring, cite it with the symbol H, always preceded by the ring locant of that carbon atom. Unsaturations in such a ring are NOT cited.

(b) If there are TWO OR MORE such carbon atoms in any ring, show this with the symbol T without a ring locant.

(c) If one or more unsaturations are present in a ring with a T saturation sign, cite them in locant path order with the appropriate U or UU symbols, preceded by the ring locant which immediately precedes the unsaturation in moving along the locant path. Omit an "a" locant only in carbocyclic monocycles. (If the unsaturation is between nonconsecutive locants in a fused or perifused ring junction, cite the earlier locant to which it is attached, then the U symbol, then a hyphen and then the later locant to which it is attached, citing such unsaturations in the order of the earlier locant.) Cite all U or UU symbols AFTER citing other cited ring segments and BEFORE citing H or T ring saturation symbols. Do NOT cite unsaturations that are fully shared with a ring not having a T saturation sign.

(d) If more than one ring is present, cite for each ring numeral a T or & symbol, where & means that the ring does not have a T saturation sign; cite these in the same order as their corresponding ring numerals, but without ring locants. If ALL the rings are cited with T, cite only a single T before the closing J. If ALL the rings are cited

with &, omit ALL the U, &, and T symbols entirely, citing only H symbols when required.

Rule 33. Within ring sign use the Y symbol only and always to designate a cyclic carbon atom to which any ring substituent (except oxygen) is doubly bonded.

Rule 34. Cite any H and W symbols inside the ring signs immediately after the symbols of the heterocyclic atoms to which they are attached.

Rule 35. Within ring signs enclose any numeral larger than nine within hyphens.

Rule 36. (a) Reserve the locants "X", "Y", and "Z" for denoting unknown or uncertain positions of substitution. Distinguish locants beyond the 23rd (W) by adding an ampersand for the first repetition of the alphabet, two ampersands for the second repetition, etc.: A& for the 24th locant, B& for the 25th, W& for the 46th, A&& for the 47th, etc.

(b) When an ampersand required by other rules immediately precedes a ring saturation & or T

sign, interpose a hyphen before the first saturation sign.

Rule 37. When a ring locant chain must branch in two directions from one ring locant in order to include all the ring atoms, designate the locant(s) in the shorter branch with the earliest main-chain locant to which the branch is attached, suffixing a hyphen to the first locant in the branch, two hyphens to the second, etc. Rank a hyphenated locant letter between the preceding locant and the next consecutive letter when applying the various rules.

Rule 38. When a ring locant chain must branch in three directions from one ring locant, or when a chain of hyphenated locants must branch in two directions, in order to include all the ring atoms designate each chain of branch locants by the principle of rule 37, but add an & symbol after the hyphen(s) of each locant in the branch that is cited last in the notation.

The preceding rules show the constraints of the ring decoder. Every type of ring will be decoded except rings

containing multiple bridges such as the adamantane derived compound. This restriction will probably affect less than a tenth of a percent of existing compounds.

3.10 CHAINS OF RINGS OTHER THAN BENZENE

Rule 39. (a) When two or more ring systems, other than benzene, are linked to form a chain of rings, choose the ring system to cite first in the notation by the following series of choices, going on to the next choice only if the earlier one gives equal alternatives or does not apply. First denote each ring system by the preceding rules; then cite first the ring system: (1) whose notation has the most ring numerals; (2) whose notation has the latest alphanumeric rank when any H,&, and T ring saturation signs and the closing J symbol are excluded; but if equal alternatives remain, when they are included; (3) which has the fewest other ring systems directly substituted upon it, with or without intervening acyclic symbols; (4) which has the earliest alphabetic set of locants in its ring for ALL its

ring substituents, including those containing other rings; (5) Which, when encoded by rules 39b-d, will produce the notation that has, first, the earliest set of ring locants in the notation for ALL the ring systems present (including benzene rings), and after this, that has the latest alphanumeric position for all other symbols.

(b) Next cite any open chain and benzene ring substituents on the initial ring structure in ascending order of their locants in this structure, punctuating any chains containing benzene rings by rule 24. If equal alternatives exist for assigning locants in the initial ring, resolve them by rule 2 (i.e., assign the earliest locant in the initial ring to the substituent whose notation symbols have latest alphanumeric position).

(c) Next cite, by step (d), the substituent chain containing the FEWEST RING SYSTEMS, citing any that contain the same number of ring systems in ascending alphabetic order of their ring locants in the initial ring system. Repeat this process for the substituent chains with the next fewest

ring systems, etc. If at any point in this process, equal alternatives exist for assigning ring locants in the initial ring structure, resolve them by first applying rule 2 to the substituent chain leading to the following ring, and after this, by applying rules 39a-1 through 39a-4 to determine which of the following rings to cite first; the first cited is assigned the earliest locant in the initial ring system. Resolve any equal alternatives for citing rings at branch points in substituents by rules 39a-1 through 39a-4. Punctuate chains containing rings by rule 24.

(d) Cite completely, including branches, any symbol chain leading to the following ring before citing this ring; denote the point of attachment on this ring by a prefixed locant preceded by a hyphen. Resolve any equal alternatives for assigning locants first by rule 30n and then by rule 2.

There are no special comments on these types of chains except back and forth use of chain and ring module.

3.11 MULTIPLIERS WITH CYCLICS NOTATIONS

Rule 40. (a) Extend rule 25 to the multiplication of symbols of substituents on rings whose notations begin with L or T, but ONLY if NO branch ring locants are present. Do not omit an "a" locant, either before or after the ring signs. Cite all ring substituents that do not lead to other L or T ring notations in ascending alphabetic order of their ring locants. Preserve this order by contracting ONLY ONE SET of identical substituents following a ring notation, choosing the set that saves the most marks, resolving equal alternatives by rule 2. Place hyphens after the locants of the multiplied groups as required by rule 26, and cite the slash mark, the multiplied symbols, and the multiplier immediately after the last such hyphenated locant.

(b) Extend rule 27 to the multiplication of symbols of substituents on rings whose notations began with L or T, but ONLY if ALL the ring positions

are filled with IDENTICAL substituents; replace all the ring locants of the substituents with a single hyphen joined to the initial L or T or the closing J symbol of their ring description.

Rule 41. (a) Within ring signs use X as a cited ring segment symbol only to denote acyclic carbon atom that is bonded to four other ring atoms in the same ring system.

(b) Add an ampersand after any cited ring segment symbol except X if the symbol is bonded to four other ring atoms in the same ring system and is not in a simple spiro ring junction indicated with the -& signs (rule 42). Place any H or W symbol attached to the spiro atom after the & sign. In order to provide a simple, direct means of denoting ring positions in simple spiro structures, they are denoted by treating them as a chain of rings according to the following rule.

Multipliers are not implemented for reasons to be explained later.

3.12 SPIRO RINGS

Rule 42. When a spiro atom that is not part of perifused ring junction is the only link between two or more rings, denote the structure as a chain of rings, following rule 39, by citing the spiro atom and any required &, H, or W symbols attached to it, with appropriate locants, in the notations for EACH of the connected rings, and adding an ampersand as a spiro junction sign after the hyphen joining the first ring notation to the locant of the next ring notation. This -& sign shows that the two ring notations it connects are part of a single ring system and that the symbols at the linked positions in each of the ring notations represent a SINGLE ATOM through which the rings are joined. Similarly, any H symbol cited after each spiro atom represents a SINGLE hydrogen atom in the structure; a W symbol, so used, represents only two oxygen atoms.

3.13 RINGS WITH PSEUDO BRIDGES

Rule 43. If three cited rings share the same pair of multicyclic points,

(a) list all the pairs of those connected ring locants that are not consecutive letters in the alphabet, citing the earlier locant first in each pair.

(b) Separately, list all the ring fusion locants to be cited in the notation, including all the implied "a" locants.

(c) For each ring fusion locant in list (b), delete from list (a) ONE pair of nonconsecutive locants having that ring fusion locant as its initial letter, UNLESS two or more locant pairs in list (a) have the same initial letter and this letter occurs fewer times in list (b), in which case do NOT delete any of these locant pairs from list (a).

(d) Cite by rule 31c any nonconsecutive locant pair(s) remaining in list (a) after these deletions.

This rule presents a real problem for the decoding program because it is the only case where a "J" which is used as ring terminator if not preceded by a space, may occur without a preceding space to identify a locant. The decoding process must recognize the presence of a "/" not yet satisfied by two locants which may consist of several letters.

3.14 RINGS OF RINGS CONTRACTION

Rule 44. Denote as a ring of rings contraction any structure in which a symbol chain containing one or more rings, with or without other symbols, is joined at both ends to a bicyclic or larger ring system which itself does not qualify for this contraction, IF either of the following is the case: (1) The symbol chain closes a single ring of ten or more atoms AND the preceding rules would require one or more branch locants to express the notation for any ring in the symbol chain; OR (2) branch locants would be required by the preceding rules for All the ring positions in any one ring.

Rule 45. Use the symbol zero, 0, as a ring locant to denote that the symbols that follow it are not attached to the preceding ring through a definite ring position (as in metallocenes and catenanes).

Rule 46. (a) Use the symbol zero, 0, as a ring saturation sign for any ring that is pi-bonded to a metal atom.

(b) Denote a simple benzene ring by L60J when it is pi-bonded to a metal atom.

Rule 47. Extend rule 44 (the ring of rings contraction) to include central rings of any size that are formed by closing one or more chains of pi-bonded rings on a monocyclic or larger structure. Count a ring that is pi-bonded to a metal atom as a single atomic unit in the size of a ring of pi-bonded rings.

This subject is not implemented for reasons explained later.

CHAPTER 4

4.1 PROGRAM CHARACTERISTICS

The program submitted as this thesis body is a good compromise between a complete system and an impractically large program which would be reserved to an extremely big comouter.

It may become apparent that with an increasing demand of decoder generality, the system core memory requirement grows without bounds, consequently it becomes necessary to choose a reasonable limit on generality which is compatible with computing practices.

The first limitation will reject all kinds of multipliers. The practice of this type of data reduction in WLN is contested by many users. Farell, C.,D. (15) says that contracted WLN strings are less desirable for creating structural diagrams and he judges that the slight total saving in symbols does not appear to justify the use of multipliers in view of the problems they present in computer-based systems. Furthermore G. A. Miller in his decoding system(20) refuses to consider the use of multipliers. And finally I.S.I(11) does not use multipliers

in the Index Chemicus which provides tapes of around twenty thousand new compounds coded in WLN per month.

The second restriction is that, the logic used cannot decode compounds with multiple bridges such as the adamantane type. The program uses consecutive entry in the connectivity table for all locants of a ring definition. When, as described by rule 37 the use of the same locant symbol is followed by one or two hypens it is necessary to use of secondary connectivity table and series of pointers. The number of tables will increase the size of the ring decoder by a factor of two for a marginally useful expanse in generality when this type of compound covers less than a tenth of a percent of the chemical literature.

4.2 SYSTEM PERFORMANCE

The Wiswesser decoder and display programm is written in FORTRAN IV on a XEROX Sigma9 computer using the Tektronix 4015 type of interactive terminal. The XEROX FORTRAN IV is more powerful than the standard FORTRAN and for this reason the program may need some slight modifications to run on other hardware. The library calls for graphic display on the Tektronix terminal is limited to a MOVE and a DRAW

function and may be easily modified for any kind of display device.

4.2.1 PROGRAM DIMENSION

Module	Size	Comments
MAIN program	67 (words)	Supervise the execution of the different modules.
CLEAR	69 (words)	Initiate different tables and stacks.
ENTER	111 (words)	Input and mapping module.
RINGDEF	286 (words)	Scan the ring part of the code and store information in different stcks.
ILOC	41 (words)	Function to compute a locant index.
RGANAL	495 (words)	Provide ring information in the connectivity table for previously composed stacks.
SKIP	18 (words)	Function to provide a right shift in connectivity table for multiple entries.
ADJ	25 (words)	Function to adjust angles between 0 and 360 degrees.

RGSAN	309 (words)	Provide entries in connectivity table for heteroelements found in ring definition code.
RGPS	238 (words)	Create the drawing table from the connectivity table.
DRAG	90 (words)	Display the drawing table on the graphic screen.
CHAIN	1158 (words)	Analyse chain codes and provide information to the connectivity table.
ICH	35 (words)	Compute the orientation to be taken for an element bonded to more than two other elements.
BOX	189 (words)	Compute the bounds of an imaginary box around string of characters.
FRAME	90 (words)	Compute from the drawing table the virtual screen to frame the graphic picture.
COMMON DATA BLOCK	4943 (words)	Space required by the different tables used in common.

The load module created on the XEROX computer occupies 9K words of core memory. A word on that system contains 32 bits.

The cpu time required for the display of one chemical structure varies from 0.3 to 1.2 seconds on our system which is in the same range as C. D. Farell reports(15) for his system running a PDP 10 computer.

4.2.2 ILLUSTRATIONS

WLN CODE
GXGGX1&2&OYO2&OX1&1&XGGG

CPU TIME SPENT = .81 SEC.

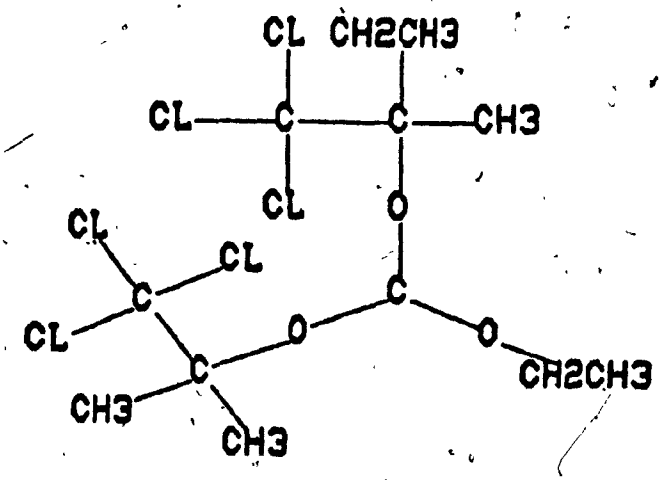


FIG 1. BRANCHING CHAIN

WLN CODE
Z1YVQM1VMY1VQ1VQ

CPU TIME SPENT = .46 SEC.

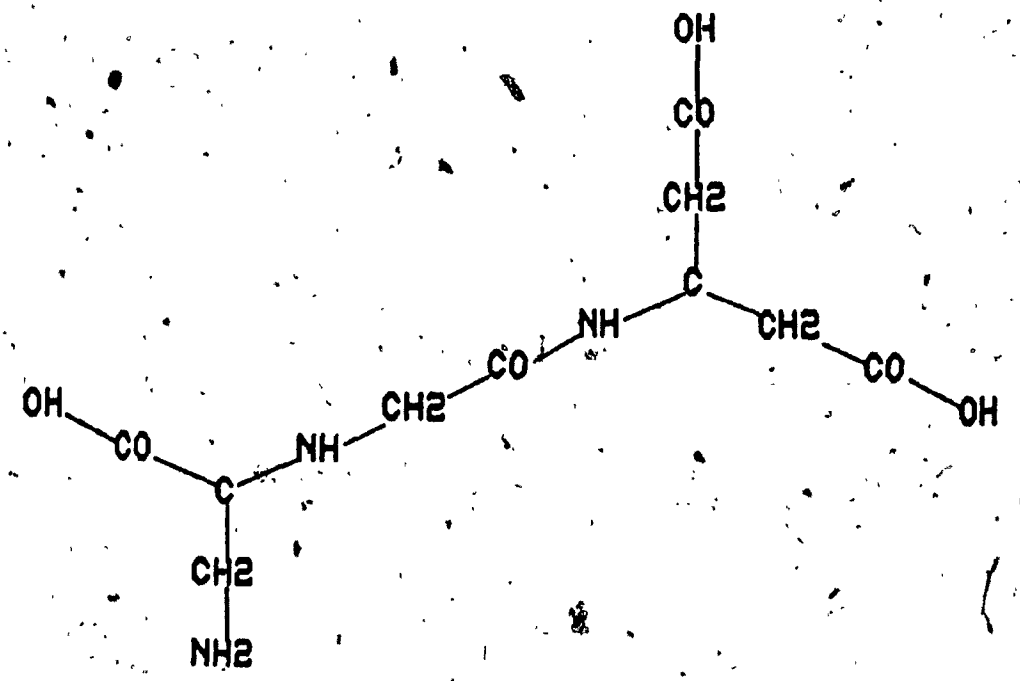


FIG 2. BRANCHING CHAIN

WLN CODE

ZR CYR CE&3MWR

CPU TIME SPENT = .81 SEC.

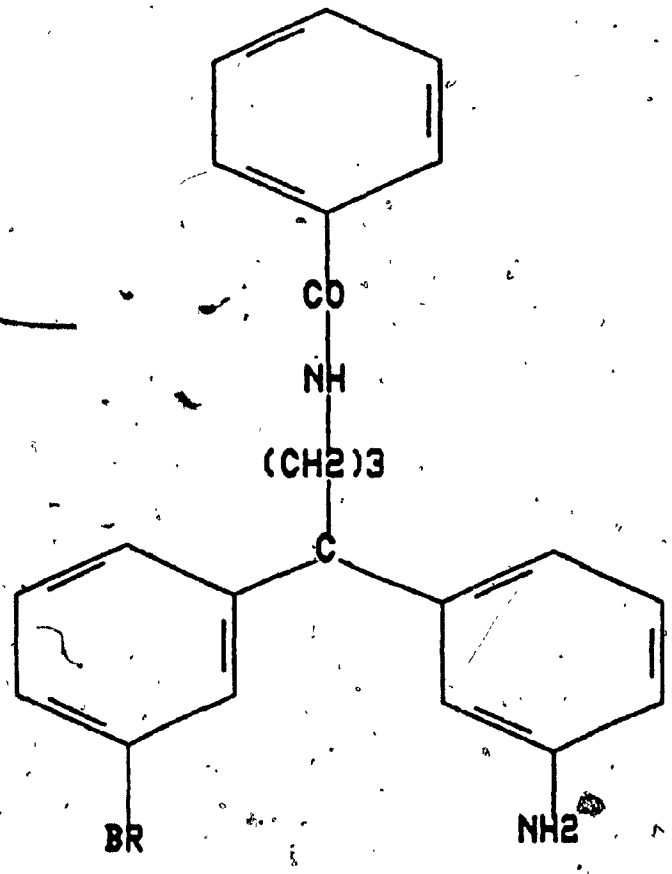


FIG 3. CHAIN OF BENZENE RINGS

WLN CODE
G1W1&M1VR CV1MR DG& DN1&1

CPU TIME SPENT = .84 SEC.

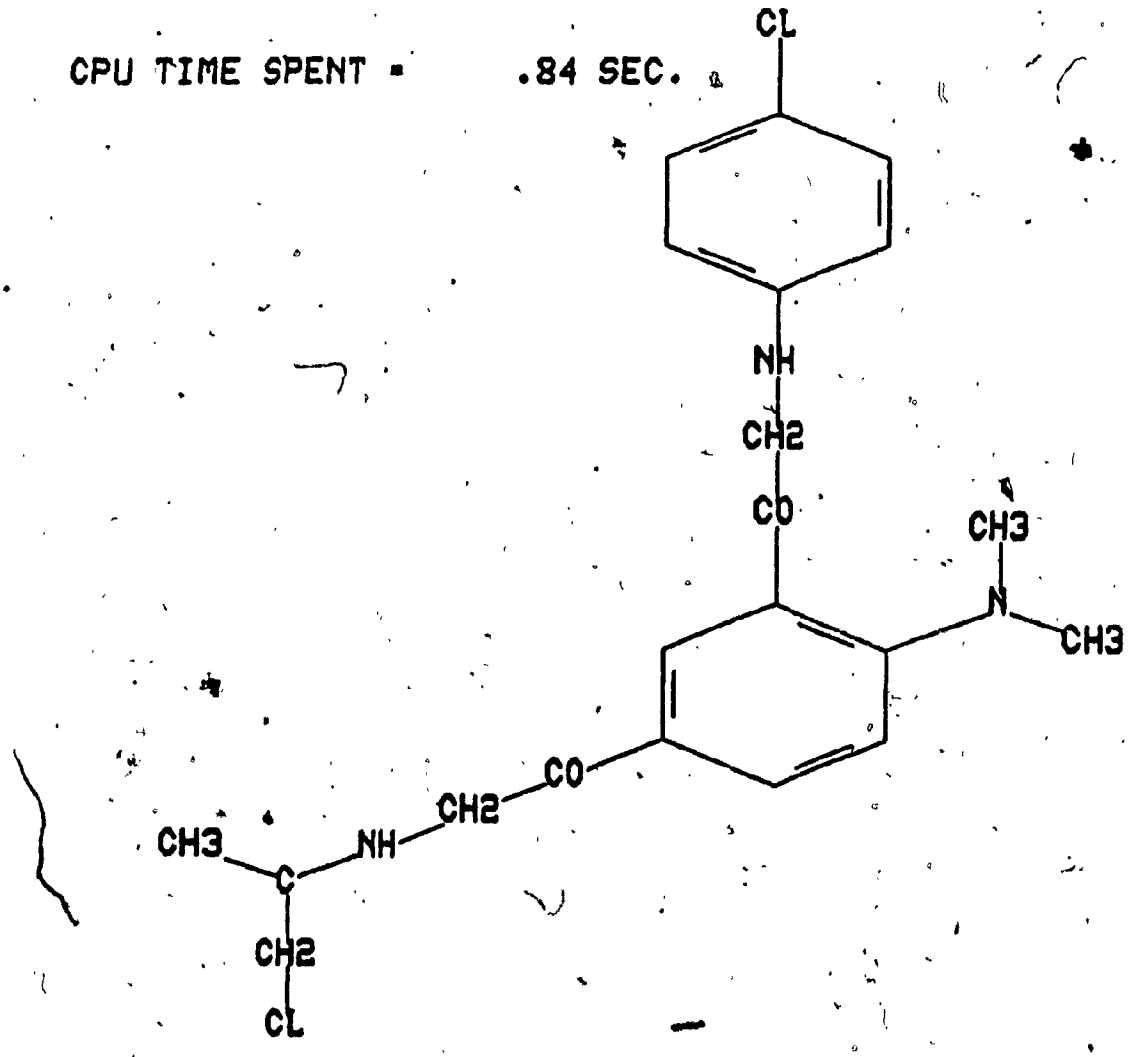


FIG 4. CHAIN OF BENZENE RINGS

WLN CODE
2N2&R COVMR D01

CPU TIME SPENT = .46 SEC.

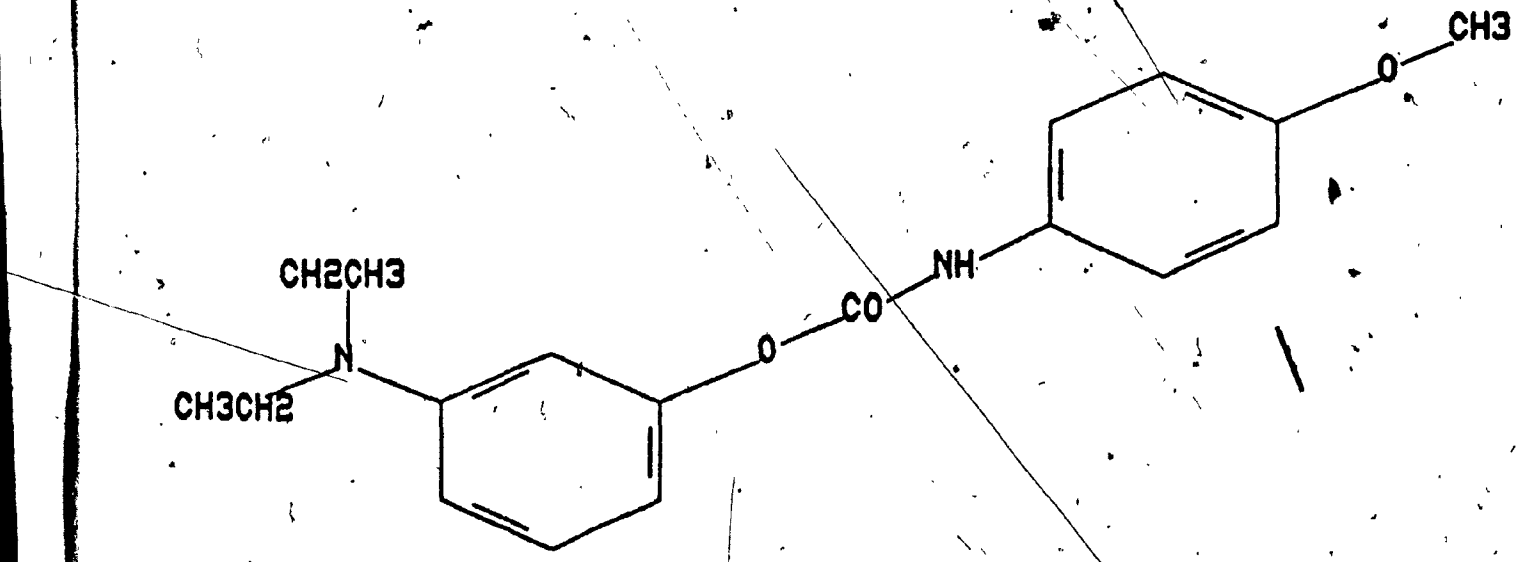


FIG 5. CHAIN OF BENZENE RINGS

WLN CODE
T5NYMV EHJ A1 BUM E1R

CPU TIME SPENT = .44 SEC.

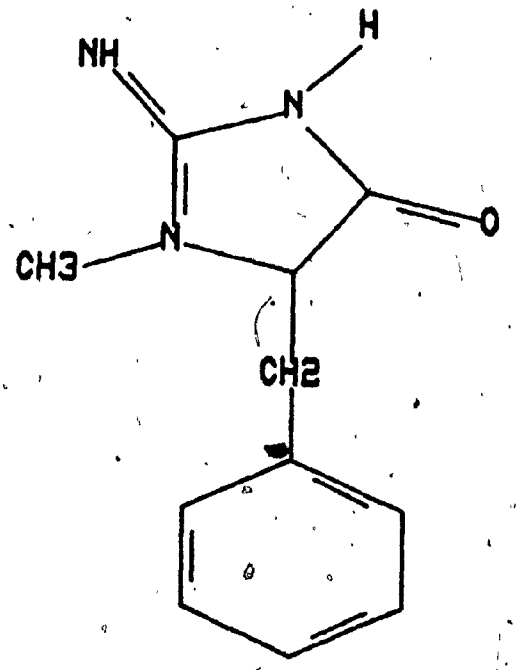


FIG 6. CYCLIC RING OTHER THAN BENZENE

WLN CODE
T5M CN BUTJ B1NR&1R

CPU TIME SPENT = .46 SEC.

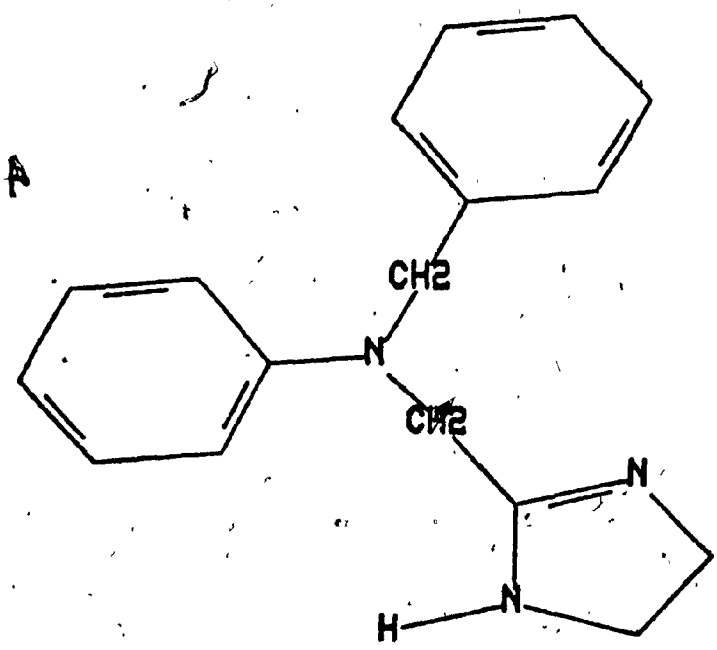


FIG 7. BRANCH OF BENZENE AND OTHER RING

WLN CODE
L E5 B666 MUTJ A E FV1Q FQ OQ

CPU TIME SPENT = .85 SEC.

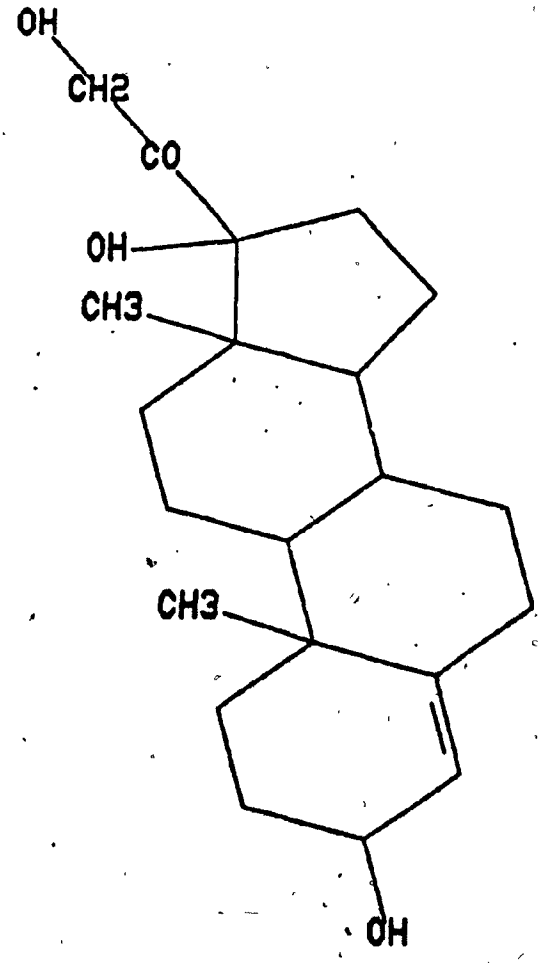


FIG 8. POLYFUSED RING STRUCTURE

WLN CODE
T F5 D6 B656 CN GO IO OM HHJ

CPU TIME SPENT = .82 SEC.

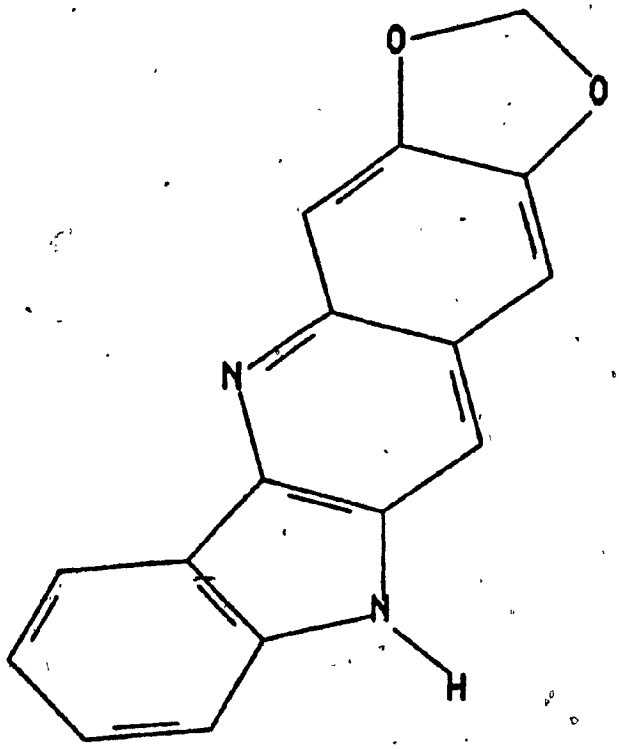


FIG 9. POLYFUSED RING WITH HETEROELEMENTS

WLN CODE

T B6 H676 COJ

CPU TIME SPENT = .44 SEC.

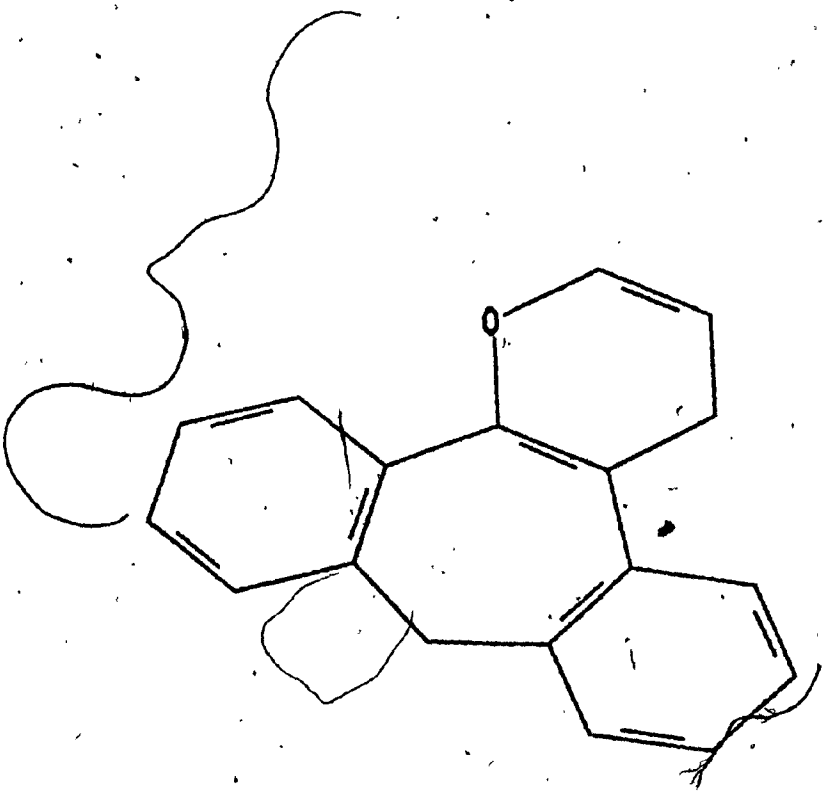


FIG 10. POLYFUSED RING

WLN CODE
T D6 B656 LMJ CQ F01 G01

CPU TIME SPENT = .81 SEC.

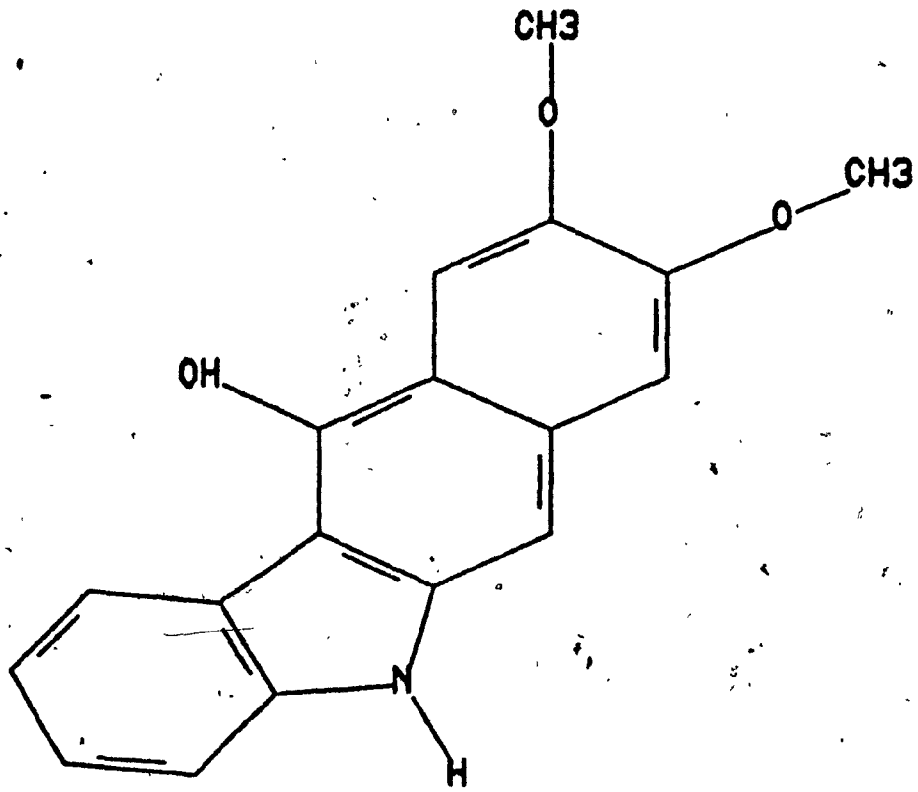


FIG 11 POLYFUSED RING WITH SUBSTITUENTS

WLN CODE
L666 B6 C6 3ABC S EHJ

CPU TIME SPENT = .49 SEC.

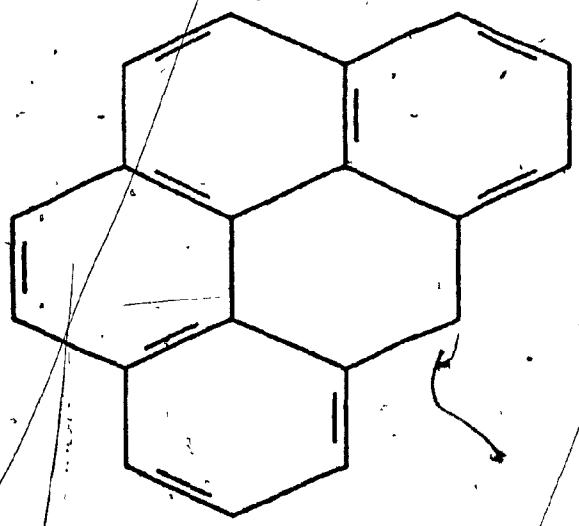


FIG 12. PERIFUSED RING

WLN CODE
T F6 D5 C666 EM ON&&TTTJ HO1 TO1 UVO1 SO- BT6OTJ

CPU TIME SPENT = 1.22 SEC.

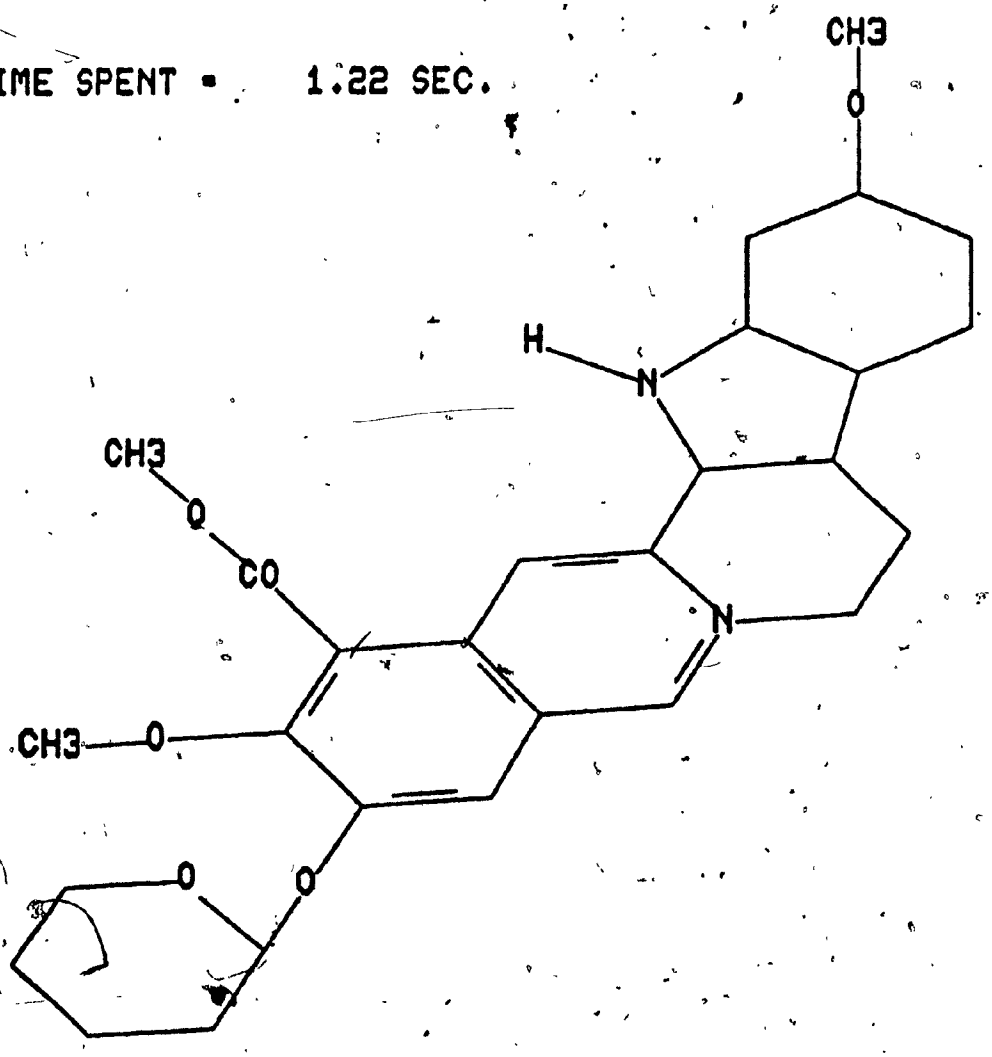


FIG 13. CHAIN OF POLYFUSED RINGS

WLN CODE
T D66 K666 1A U CS JN QVJ

CPU TIME SPENT = .82 SEC.

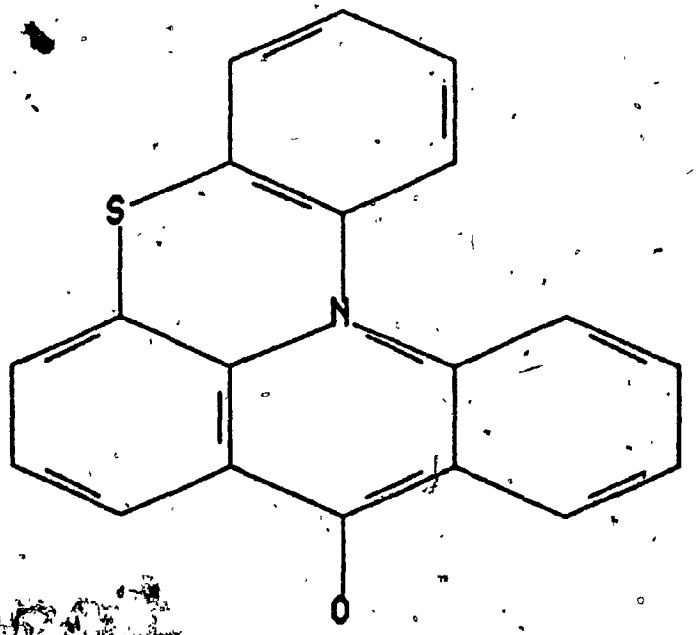


FIG. 14. PERIFUSED RING WITH HETEROELEMENTS

WLN CODE
T6NJ B02N2&2 FXQR DR&&- BT5OJ

CPU TIME SPENT = .89 SEC.

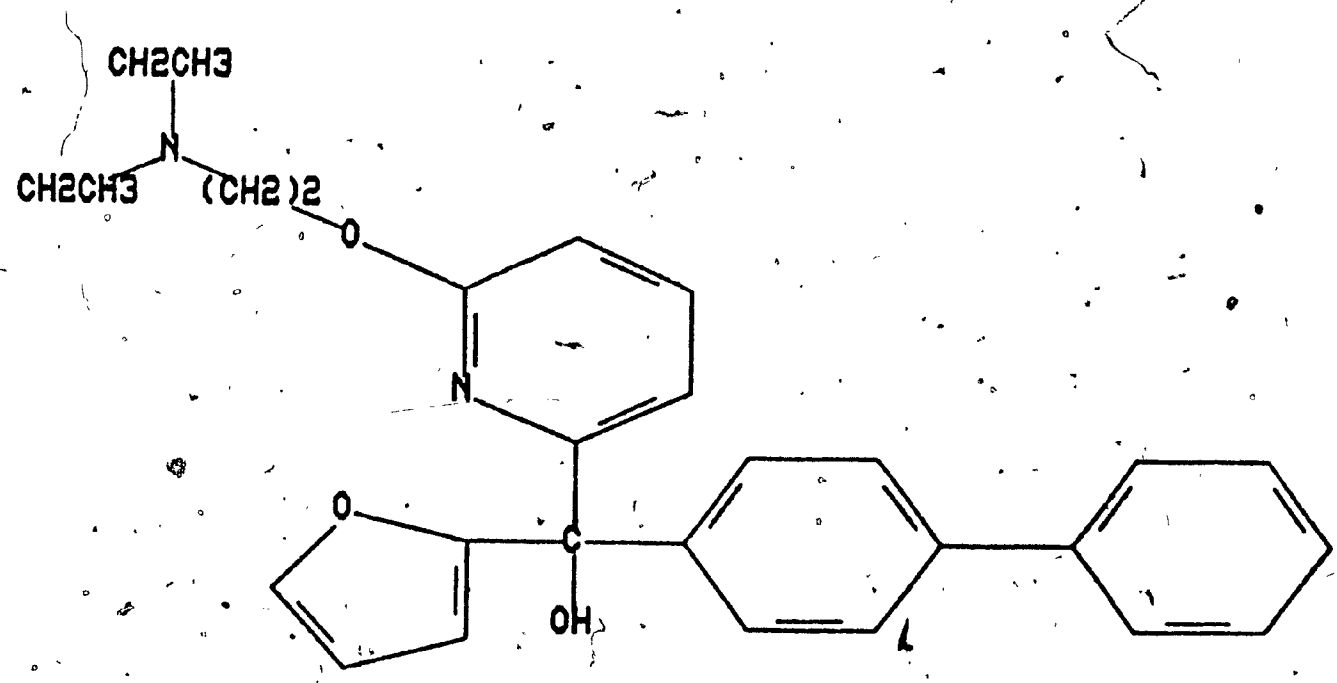


FIG 15. BRANCHING CHAIN OF BENZENE AND HETEROCYCLES.

WLN CODE
T6NJ BN2N1&1&1- BT5SJ FE

CPU TIME SPENT = .80 SEC.

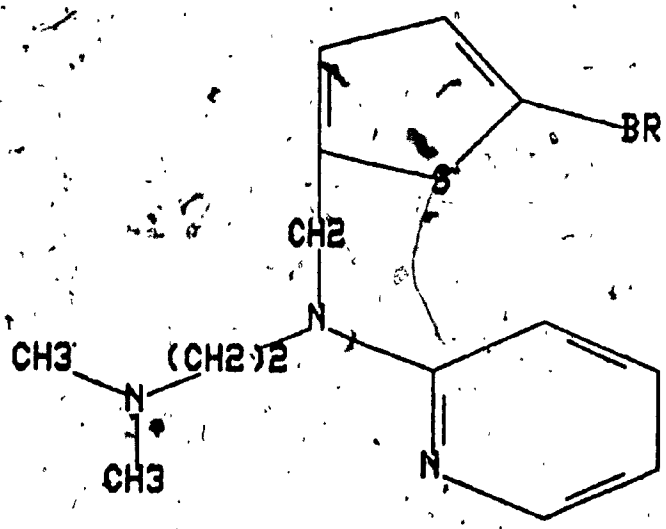


FIG 16. BRANCH OF CHAIN AND HETEROCYCLES

WLN CODE
T6B CNJ H- HT66 CNJ D- AL6TJ C- AL6TJ

CPU TIME SPENT = .86 SEC.

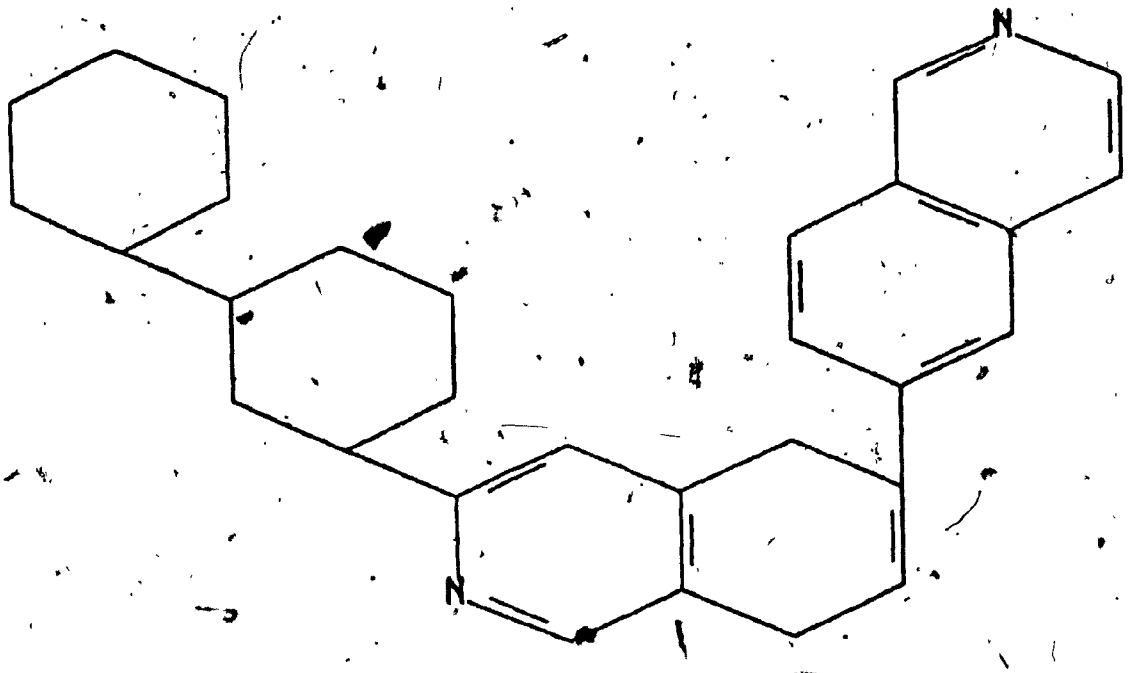


FIG 17. CHAIN OF BENZENE AND POLYFUSED RINGS

REFERENCES

1. Wiswesser, W. J., "107 Years of Line-Formula Notations (1861-1968)", J. Chem. Doc., 8, 146, (1968).
2. Mitchell, A. D., "British Chemical Nomenclature", p. 2, E. Arnold & Co., London.
3. Erlenmeyer, E., "Ueber Propylverbindungen", Ann. Chemie, 126, 306, (1863).
4. Kekule, A., "Untersuchungen über aromatische Verbindungen", Ann. Chemie, 137, 130-96, (1866).
5. Gordon, M., Kendall, C. E., and Davidson, W. H. T., "A New Systematisation of Chemical Species", Proc. 11th Int. Congr. Pure & Appl. Chem., London, Vol 2, Sec 3, p. 115-32, July 1947.
6. Davidson, W. H. T., and Gordon, M., "Sorting for Chemical Groups using Gordon-Kendall-Davidson Ciphers", Ann. Doc., 8, 202-10, (1957).
7. NAS-NRC, Survey of Chemical Notation Systems, Publ. 1150, p. 8, Washington, D.C., (1964).
8. Zipf, G. K., "Human Nature and the principle of Least Effort", Addison-Wesley Press Inc., Cambridge, (1949).
9. Wiswesser, W. J., "A Line-Formula Chemical Notation", pp. 138-40, Crowell Co, New York, (1954).
10. Smith, E. G., "The Wiswesser Line-Formula Chemical

Notation", McGrawn-Hill, New York, (1968).

11. Institute for Scientific Information, Index Chemicus Registry System (ICRS), February 1968 announcement.

12. Hyde, E., and Thomson, L., "Structure Display", J. Chem. Doc., 8, 138-46, (1968).

13. Hyde, E., Matthews, F.W., Thomson, L.H., and Wiswesser, W.J., "Conversion of Wiswesser Notation to a Connectivity Matrix for Organic Compounds", J. Chem. Doc., 7, 200-4, (1967).

14. Gottardi, R., "A Modified Dot-Bond Structural Formula Front with Improved Stereochemical Notation Abilities", J. Chem. Doc., 10, 75-81, (1970).

15. Farell, C.D., Chauvenet, A.R., and Koniver, D.A., "Computer Generation of Wiswesser Line Notation", J. Chem. Doc., 11, 52-9, (1971).

16. Rogers, M.A., "Organic Chemical Structures on Computers", Chemistry and Industry, pp.952-4, (1970).

17. Feldman, R.J., and Koniver, D.A., "Interactive Searching of Chemical Files and Structural Diagram Generation from Wiswesser Line Notation", J. Chem. Doc., 11, 154-9, (1971).

18. Feldman, R.J., Heller, S.R., and Shapiro, K.P., "An Application of Interactive Computing - A Chemical Information System", J. Chem. Doc., 12, 41-7, (1972).

19. Feldman, R.J., and Heller, S.R., "An Application of Interactive Graphics - The Nested Retrieval of Chemical Structures", J. Chem. Doc., 12, 48-54, (1972).

20. Miller, G.A., "Encoding and Decoding WLN", J. Chem. Doc., 12, 60-67, (1972).

21. Feldman A., "A Chemical Teletype", J. Chem. Doc., 13, 53-56, (1973).

22. Leo, A., Elkins, D., and Hansch, C., "Computerized Management of Structure-Activity Data. II. Decoding and Searching Branching Chains and Multiplied Groups Coded in WLN", J. Chem. Doc., 14, 61-65, (1974).

23. Elkins, D., Leo, A., and Hansch, C., "Computerized Management of Structure-Activity Data. III. Computerized Decoding and Manipulation of Ring Structures Coded in WLN", J. Chem. Doc., 14, 65-69, (1974).

APPENDIX A PROGRAM LISTING

```
      INTEGER FORM,SYM
      COMMON /A/ FORM(80),SYM(40),NEND
C*************************************************************
C     FORM = CARD IMAGE
C     SYM = SYMBOL VALID SEE DATA STATEMENT
C     NEND = FLAG FOR EOF
C*************************************************************
      COMMON /B/ IBND, IFOR(80),ITA(40),IFL(20)
C*************************************************************
C     IFOR = CONVERTED CARD IMAGE
C     ITA= OUTPUT SYMBOL
C     IFL = FLAG ARRAY
C     IFL(1)=NUMBER OF RINGS IN CYCLE NOTATION
C     IFL(2)=POSITION OF CHARACTER IN FORMULA TO BE ANALYSED
C     IFL(3)=NUMBER OF CHARACTERS IN FORMULA NOTATION
C     IFL(4)=TYPE OF CYCLE TO BE ANALYZED :
C                  1 FOR HETEROCYCLE
C                  0 FOR CARBOHYDRATE
C                 -1 FOR CHELATE
C     IFL(5)=POSITION OF END OF RING IN NOTATION
C     IFL(6)=POSITION OF RELATIVE START OF LABEL IN CONNECTING TABLE
C     IFL(7)=FLAG FOR TYPE OF RING STRUCTURE
C                  0 NORMAL POLYCYCLIC FUSED RING
C                  N NUMBER OF CENTER ATOMS FOR PERI-FUSED RING.
C     IFL(8)=LAST ENTRY IN TABLE AT
C     IFL(9)=SPIRO NUMBER
C     IFL(10)=NEXT RING LOCATION
C     IFL(11)=PREVIOUS RING LOCATION
C     IFL(12)=FLAG FOR P REDUCTION (RULE 8C)
C     IFL(13)=FLAG FOR MULTIPLIER
```

```
*     IFL(14)=NUMBER OF BRIDGED ELEMENTS
*     IFL(15)=POINTER FOR LAST ENTRY IN ICHAR
*     IFL(16)= NUMBER OF SPECIAL LINKS LISTED IN KLINK
*     IFL(17)=END OF RING SYSTEM EXCLUDING SATURATION SYMBOLS
*     IFL(18)=LAST LOCANT LABEL FOR RING STRUCTURE
*     IFL(19)=NUMBER OF ENTRIES IN DRW TABLE
*     IFL(20)=FLAG FOR ERROR AND RETURN IN QUERY
*****************************************************************
      COMMON /C/ AT(150,20),DRW(250,4),KRG(20,3),ICHAR(200),KPER(20)
*****************************************************************
*     AT= CONNECTING TABLE
*     AT(-,1)= POINTER FOR NEXT LOCATION TO BE USED
*     AT(-,2)= X COORDINATE OF THE ELEMENT
*     AT(-,3)= Y      "              "
*     AT(-,4)= POINTER OF LOCATION TO JOIN FIRST PAIR
*     AT(-,5)= ABSOLUTE ANGLE TO JOIN THE LOCATION FIRST PAIR
*     AT(-,6)= SAME AS 4  SECOND PAIR
*     AT(-,7)= SAME AS 5  SECOND PAIR
*     AT(-,8)= SAME AS 4 THIRD PAIR
*     AT(-,9)= SAME AS 5  THIRD PAIR
*     AT(-,10)= SAME AS 4 FOURTH PAIR
*     AT(-,11)= SAME AS 5  FOURTH PAIR
*     AT(-,12)= FLAG FOR UNSATURATION
*     AT(-,13)= NUMBER OF BOND USED OUTSIDE
*     AT(-,14)= DIRECTION TO BE USED FOR OUTSIDE OF RING
*     AT(-,15)= NUMBER OF CHARACTERS TO REPRESENT THE ELEMENT
*     AT(-,16)= POSITION OF THE FIRST CHARACTER IN ICHAR
*     AT(-,17)= FLAG FOR DIFFERENTIATE CHAIN/RING BOND
*     AT(-,18)= FLAG FOR SPIRO CONNEXTION
*     AT(-,19)= FLAG FOR MULTIPLE ENTRY USED IN ENTRX
*     AT(-,20)= FLAG TO COUNT ATOM USAGE FOR MULTIPLE RING
*     DRW = DRAWING TABLE
*     DRW IS BUILD SO THAT THE DISPLAY PROCESSOR HAS
```

```
*     TO MOVE TO DRW(1),DRW(2) THEN DRAW TO DRW(3),DRW(4).
*     KRG = RING INFORMATION
*           KRG(-,1) = EARLY LOCANT
*           KRG(-,2) = RING SIZE
*           KRG(-,3) = RING SATURATION 1 SAT, 2 UNSAT
*     ICHAR= VECTOR OF NUMBER REPRESENTATION OF THE CHARACTER TO
*     BE DISPLAYED AT THE ELEMENT POINTER IN AT(-,15) AND AT(-,16).
*     KPER= VECTOR OF ELEMENT IN A BRIDGE (THERE ARE IFL(7)
*     ENTRIES IN KPER) THE ELEMENTS SUFFIXED BY - ARE INCREASED BY 1000
*     AND ELEMENTS SUFFIXED BY & ARE INCREASED BY 100.
*************************************************************
      COMMON /D/ ISAC(40),IST,KLINK(20,2),KBRG(20)
*************************************************************
*     ISAC = STACK OF POINTER. IST POINTER IN STACK ISAC
*     KLINK= STACK OF SPECIAL LINKS REPRESENTED WITHIN A RING DEFINITION
*     AS /PP
*************************************************************
      DATA SYM/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H ,1H&,1H-,1H/,
     11HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO,1HP,
     21HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ/
      DATA ITA/48,49,50,51,52,53,54,55,56,57,40,41,43,45,
     165,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
     281,82,83,84,85,86,87,88,89,90/
*************************************************************
*     INITIALIZATION OF ALL VARIABLES
*     ENTER A FORMULA NOTATION IN ASCII CHARACTERS CHECK FOR VALIDITY
*     AND CODE IN INTEGER AS FOLLOW : DIGITS (0-9), LETTERS (14-39),
*     SPECIAL SYMBOLS  SP,&,-,/ (10-13)
*     CHECK OF END-OF-FILE BY NEND=1
*     BRANCH BETWEEN CHAIN DECODING OR RING DECODING
*     AT END BUILD THE DRAWING TABLE
*     AND DISPLAY THE STRUCTURE
*************************************************************
```

```
X     GOTO 1
      CALL INITT (240)
      CALL ANMODE
1     CALL CLEAR
      CALL ENTER
      IF(NEND.EQ.1)STOP
2     I=IFL(2)+1
      IF(IFOR(I).EQ.33.OR.IFOR(I).EQ.25.OR.IFOR(I).EQ.17)GOTO 3
      CALL CHAIN
      IF(IFL(20).NE.0)GOTO 1
      IF(IFL(2).LT.IFL(3))GOTO 2
      GOTO 4
3     CALL RINGDEF
      IF(IFL(20).NE.0)GOTO 1
      CALL RGANAL
      IF(IFL(20).NE.0)GOTO 1
      CALL RGSAN
      IF(IFL(20).NE.0)GOTO 1
      IF(IFL(2).LT.IFL(3))GOTO 2
4     CALL RGPS
      IF(IFL(20).NE.0)GOTO 1
      CALL FRAME
X     GOTO 1
      CALL DRAG
      CALL TINPUT(I)
      CALL ERASE
      GOTO 1
      END
```

```
      SUBROUTINE CLEAR
      INTEGER FORM,SYM
      COMMON /A/ FORM(80),SYM(40),NEND
      COMMON /B/ IBND, IFOR(80),ITA(40),IFL(20)
      COMMON /C/ AT(150,20),DRW(250,4),KRG(20,3),ICHAR(200),KPER(20)
      COMMON /D/ ISAC(40),IST,KLINK(20,2),KBRG(20)
*************************************************************
*     SET THE NEXT POINTER IN TABLE AT
*     CLEAR THE OUTSIDE DIRECTION BY A NEGATIVE NUMBER
*************************************************************
      DO 1 I=1,150
      AT(I,1)=I+1
      DO 1 J=2,20
1     AT(I,J)=0
      DO 2 I=1,150
2     AT(I,14)=-1
      DO 3 I=1,250
      DO 3 J=1,4
3     DRW(I,J)=0.
      DO 4 I=1,20
4     IFL(I)=KRG(I,1)=0
      NEND=IBND=0
      IST=0
      DO 5 I=1,40
5     ISAC(I)=0
      RETURN
      END
```

```
      SUBROUTINE ENTER
      INTEGER FORM,SYM
      COMMON /A/ FORM(80),SYM(40),NEND
      COMMON /B/ IBND, IFOR(80),ITA(40),IFL(20)
      COMMON /C/ AT(150,20),DRW(250,4),KRG(20,3),ICHAR(200),KPER(20)
*****************************************************************
*     PROMPT AND READ A LINE OF CODE
*     IF THE LINE IS EMPTY STOP THE PROGRAM
*     MAP THE VECTOR READ USING DATA ARRAY SYM
*****************************************************************
1     PRINT 100
      READ(105,101)FORM
      PRINT 102,FORM
      IF(FORM(1).EQ.SYM(11))GOTO 6
      DO 4 I=1,80
      IF(FORM(I).EQ.SYM(11))GOTO 3
      DO 2 J=1,40
      IF(FORM(I).NE.SYM(J))GOTO 2
      IFOR(I)=J-1
      GOTO 4
2     CONTINUE
      PRINT 103,FORM
      GOTO 1
3     IF(FORM(I-1).EQ.SYM(11))GOTO 5
      IFOR(I)=10
4     CONTINUE
      IFL(3)=80
      RETURN
5     IFL(3)=I-2
X     PRINT 104,(IFOR(I),I=1,IFL(3))
X     PRINT 105,IFL(3)
      RETURN
6     NEND=1
```

```
      RETURN
100   FORMAT(' ENTER A WISWESSER LINE FORMULA TO BE DISPLAYED')
101   FORMAT(80A1)
102   FORMAT(1X,80A1)
103   FORMAT(1X,'THERE IS A ILLEGAL SYMBOL PRESENT DETECTED BY ENTER',
     1/1X,80A1)
X104  FORMAT(' END OF SUBROUTINE ENTER THE FORMULA IS ACCEPTED FOR '
X     1'INVALID CHARACTERS SCANNING'/' AND IS REPRESENTED IN MEMORY'
X     2' AS:'/40I3/40I3)
X105  FORMAT(' NUMBER OF CHARACTERS IN FORMULA :',I3)
      END
```

```
      SUBROUTINE RINGDEF
      COMMON /B/ IBND, IFOR(80),ITA(40),IFL(20)
      COMMON /C/ AT(150,20),DRW(250,4),KRG(20,3),ICHAR(200),KPER(20)
      COMMON /D/ ISAC(40),IST,KLINK(20,2),KBRG(20)
*************************************************************
*     IN THE TYPICAL CASE SHOWN BELOW AS EXAMPLE
*     T B666/GL A 2BG L CO GXOO DU JHT&&J
*     DETECT NUMERALS SHOWING RING SIZE PRECEDED BY THE EARLIEST
*     LOCANT IN THAT RING, BUT WITH 'A' LOCANTS OMITTED   B666
*     DETECT NONCONSECUTIVE PAIRS OF RING LOCANTS, WITHOUT SPACE
*     AND PRECEDED BY A SLASH     /GL
*     DETECT LOCANTS OF BRIDGE ATOMS IF ANY        A
*     DETECT AFTER A BLANK SPACE THE NUMBER OF MULTICYCLIC POINT LOCANTS
*     THAT FOLLOW IF ANY       2BG
*     DETECT THE LAST RING ATOM LOCANT IF MUTICYCLIC POINTS    L
*     JUMP OVER PART DECODED BY SUBROUTINE RGANAL OF SPECIAL
*     ATOMS IN THE RING     CO GXOO DU JH
*     DETECT THE SATURATION SHOWN BY T OR & AT THE END
*     IN THIS CASE             T&&J
*     SEE PROBLEM OF PREVIOUS ROUTINE WHEN TWO J'S ARE PRESENT
*************************************************************
      IST=IST+1;ISAC(IST)=IFL(8)
      LL=IFL(2)+1;LK=0;IP=1
1     LL=LL+1;IF(LL.GT.IFL(3))GOTO 22
      IF(IFOR(LL).EQ.10)GOTO 3
      IF(IFOR(LL).GT.10)GOTO 4
*************************************************************
*     A RING SIZE IS FOUND AND STORED
*************************************************************
      IX=IFOR(LL)
2     LK=LK+1
      KRG(LK,1)=IP
      KRG(LK,2)=IX
```

```
      IP=1;LX=LL
      GOTO 1
*****************************************************************
*     DETECTION OF A LOCANT
*****************************************************************
3     LL=LL+1;IF(LL.GT.IFL(3))GOTO 22
      IF(IFOR(LL).LT.14)GOTO 5
      IP=ILOC(LL)
      GOTO 1
*****************************************************************
*     CHECK FOR BIG RING AS B-12- TWO HYPHENS MUST BE PRESENT
*     DO NOT TAKE B-6 WHICH IS A LOCANT
*****************************************************************
4     IF(IFOR(LL).NE.12)GOTO 5
      IF(IFOR(LL+1).GT.9.OR.IFOR(LL+2).GT.9.OR.IFOR(LL+3).NE.12)GOTO 5
      IX=(IFOR(LL+1)*10)+IFOR(LL+2)
      LL=LL+4
      GOTO 2
*****************************************************************
*     SEARCH FOR SPECIAL LINKS
*****************************************************************
5     IF(IFOR(LX+1).NE.13)GOTO 6
      IFL(16)=IFL(16)+1;LX=LX+2
      IF(IFOR(LX).LT.14)GOTO 22
      KLINK(IFL(16),1)=ILOC(LX);LX=LX+1
      IF(IFOR(LX).LT.14)GOTO 22
      KLINK(IFL(16),2)=ILOC(LX)
      GOTO 5
*****************************************************************
*     SEARCH FOR BRIDGE NOT PERIFUSED
*****************************************************************
6     LL=LX+1
      REPEAT 7,  FOR I=(LL,IFL(3))
```

```
      IF(IFOR(I).NE.10)GOTO 8
      I=I+1
      IF(IFOR(I).LT.14)GOTO 8
      IP=ILOC(I)
      IF(IFOR(I+1).NE.10.AND.I.NE.IFL(3)-1)GOTO 8
      IFL(14)=IFL(14)+1;LX=I;KBRG(IFL(14))=IP
7     CONTINUE
*************************************************************
*     NOW DETECTECT PRESENCE OF MULTICYCLIC POINTS
*************************************************************
8     IF(IFOR(LX+1).NE.10)GOTO 10
      IF(IFOR(LX+2).GT.9)GOTO 10
      LX=LX+2;IFL(7)=IFOR(LX)
      DO 9 I=1,IFL(7)
      LX=LX+1
      KPER(I)=ILOC(LX)
9     CONTINUE
      LX=LX+1
      IF(IFOR(LX).NE.10)GOTO 22
      LX=LX+1
      IFL(18)=ILOC(LX)
*************************************************************
*     NOW DETECT THE END OF RING NOTATION BY A NON-SPACE, &
*************************************************************
10    IFL(2)=LX;IFL(1)=LK
      DO 11 I=IFL(2),IFL(3)
      IF(IFOR(I).EQ.23.AND.IFOR(I-1).NE.10)GOTO 12
11    CONTINUE
      GOTO 22
12    IFL(5)=I
*************************************************************
*     NOW DETECT SATURATION  E.G. T&&
*************************************************************
```

```
      IF((IFOR(IFL(5)-1).EQ.33.OR.IFOR(IFL(5)-1).EQ.11).AND.
     1IFL(5)-1.GT.IFL(2))GOTO 15
      IP=1;IFL(17)=IFL(5)-1
   13 DO 14 I=1,LK
   14 KRG(I,3)=IP
      GOTO 17
   15 IF(IFOR(IFL(5)-1).EQ.33.AND.(IFOR(IFL(5)-2).NE.33.AND.
     1IFOR(IFL(5)-2).NE.11))IP=0;IFL(17)=IFL(5)-2;GOTO 13
      DO 16 I=LK,1,-1
      IF(IFOR(IFL(5)-I).EQ.33)KRG(I,3)=0;GOTO 16
      IF(IFOR(IFL(5)-I).EQ.11)KRG(I,3)=1;GOTO 16
      GOTO 22
   16 CONTINUE
      IFL(17)=IFL(5)-(LK+1)
   17 CONTINUE
   X  PRINT 100
   X  DO 110 I=1,IFL(1)
   X  PRINT 101,(KRG(I,J),J=1,3)
  X110 CONTINUE
   X  IF(IFL(14).EQ.0)GOTO 112
   X  DO 111 I=1,IFL(14)
   X  PRINT 102,KBRG(I)
  X111 CONTINUE
  X112 IF(IFL(16).EQ.0)GOTO 114
   X  DO 113 I=1,IFL(16)
   X  PRINT 103,(KLINK(I,J),J=1,2)
  X113 CONTINUE
  X114 IF(IFL(7).EQ.0)RETURN
   X  DO 115 I=1,IFL(7)
   X  PRINT 104,KPER(I)
  X115 CONTINUE.
      RETURN
   22 PRINT 105
```

```
      IFL(20)=1
      RETURN
X100  FORMAT(' END OF SATUR ENTRIES IN KRG(RING DEFINITION)')
X101  FORMAT(3I3)
X102  FORMAT('BRIDGE AT',I5)
X103  FORMAT('LINK BETWEEN ',2I5)
X104  FORMAT('MULTICYCLIC POINT AT ',I5)
105   FORMAT('ERROR DETECTED BY SATUR')
      END
```

```
      FUNCTION ILOC(I)
      COMMON /B/ IBND, IFOR(80),ITA(40),IFL(20)
*****************************************************************
*     FUNCTION TO COMPUTE A LOCANT
*****************************************************************
      ILOC=IFOR(I)-13
      IF(IFOR(I+1).EQ.12)GOTO 2
1     IF(IFOR(I+1).EQ.11)I=I+1;ILOC=ILOC+23;GOTO 1
      RETURN
2     IF(IFOR(I+1).EQ.12)I=I+1;ILOC=ILOC+1000;GOTO 2
      IF(IFOR(I+1).EQ.11)I=I+1;ILOC=ILOC+100
      RETURN
      END
```

```
      SUBROUTINE RGANAL
      COMMON /B/ IBND, IFOR(80),ITA(40),IFL(20)
      COMMON /C/ AT(150,20),DRW(250,4),KRG(20,3),ICHAR(200),KPER(20)
      COMMON /D/ ISAC(40),IST,KLINK(20,2),KBRG(20)
*****************************************************************
*     ENTER THE RING IN THE CONNECTING TABLE AT
*     COMPUTE ALL ANGLE FOR BOND AS NORTH 0 DEGREES AND TURN
*     CLOCKWISE. AT THE SAME TIME COMPUTE THE OUTSIDE BOND FOR SUBSTITUENT
*     ALLOW FOR SEVERAL RINGS AND PERIFUSED RING AND BRIDGE
*****************************************************************
      IFL(6)=ISAC(IST)
      MM=0;IF=0;M=150
      DO 24 I=1,IFL(1)
      JJ=KRG(I,2)
      II=KRG(I,1)+IFL(6)
      IF(I.EQ.1)GOTO 4
      IF(AT(II,4).NE.0.AND.AT(II,4).NE.AT(II,1))CALL SKIP(II)
      AT(II,4)=AT(II,1)
*****************************************************************
*     COMPUTE THE DIRECTION FOR THE NEXT ATOM
*****************************************************************
      DO 1 I1=10,4,-2
      IF(AT(AT(II,1),I1).EQ.II)GOTO 2
1     CONTINUE
      GOTO 3
2     AT(II,5)=ADJ(AT(AT(II,1),I1+1)+180.)
      GOTO 4
3     AT(II,5)=ADJ(AT(M,5)+180-(IZ-II)*360./JJ)
      IF(M.GE.II)GOTO 4
      AT(II,5)=ADJ(AT(M,5)+(II+1-M)*360/KRG(I+1,2)+360/JJ)
      IF=KRG(I-1,2)-2
4     M=II
      DO 21 J=1,JJ
```

```
      AT(M,20)=AT(M,20)+1
      IF(AT(M,4).NE.0.AND.AT(M,4).NE.AT(M,1))CALL SKIP(M)
      AT(M,4)=AT(M,1)
      DO 5  I1=10,4,-2
      IF(AT(AT(M,1),I1).EQ.M)GOTO 6
5     CONTINUE
      GOTO 7
6     AT(M,5)=ADJ(AT(AT(M,1),I1+1)+180.)
      GOTO 8
7     AT(M,5)=ADJ(AT(II,5)+(360./JJ)*(J-1))
*****************************************************************
*     COMPUTE ANGLE FOR OUTSIDE THE RING
*****************************************************************
8     IF(AT(M,14).LT.0.)AT(M,14)=ADJ(AT(M,5)+180.+360./JJ);GOTO  9
      IF(MM.EQ.0)AT(M,14)=ADJ(AT(M,5)+180.);MM=1;GOTO  9
      AT(M,14)=ADJ(AT(M,7)+180.);MM=0
9     IF(AT(M,12).LT.1)AT(M,12)=KRG(I,3)
      IF(IFL(7).EQ.0)GOTO 15
      IF(IFL(16).EQ.0)GOTO 12
*****************************************************************
*     PROCESS FOR SPECIAL LINK
*****************************************************************
      DO 10 I1=1,IFL(16)
      IF(M.EQ.IFL(6)+KLINK(I1,2))GOTO 11
10    CONTINUE
      GOTO 12
11    AT(KLINK(I1,1)+IFL(6),1)=M
      AT(M,4)=KLINK(I1,1)+IFL(6)
      GOTO 23
*****************************************************************
*     PROCESS FOR PERIFUSED RING
*****************************************************************
12    IF(J.NE.JJ-1)GOTO 15
```

```
      DO 13 I1=1,IFL(7)
      IF(KRG(I,1).EQ.KPER(I1))GOTO 14
13    CONTINUE
      GOTO 15
14    IF(AT(II,20).LT.3)GOTO 15
      AT(M,4)=II+1
      AT(II+1,1)=M
      AT(II+1,20)=3
      GOTO 23
15    IF(IFL(14).EQ.0)GOTO 20
*****************************************************************
*     PROCESS FOR BRIDGED RING
*****************************************************************
      DO 16 I1=1,IFL(14)
      IF(KRG(I,1).EQ.KBRG(I1))GOTO 17
16    CONTINUE
      GOTO 20
17    J1=1
18    IF(IFL(14).GT.I1+J1)GOTO 19
      IF(KBRG(I1+J1).NE.KRG(I,1)+J1)GOTO 19
      J1=J1+1
      GOTO 18
19    IF(J.NE.JJ-J1)GOTO 20
      IF(AT(II,20).LT.2)GOTO 20
      AT(M,4)=II+J1
      AT(II+J1,1)=M
      AT(II+J1,20)=3
      GOTO 23
20    IF(J.EQ.JJ)GOTO 22
      M=AT(M,1)
      IF(M.GT.IFL(8))IFL(8)=M
21    CONTINUE
22    AT(II,1)=M
```

```
      AT(M,4)=II
23    IZ=II-IF
      IF=0
24    CONTINUE
*************************************************************
*     REAJUST ANGLE FOR RING AFTER OTHER RING
*************************************************************
      IF(IFL(10).EQ.0)GOTO 28
      IA=ADJ(AT(IFL(10)+IFL(6),14)+180-AT(IFL(11),14))
      DO 27 I=IFL(6)+1,IFL(8)
      DO 25 J=4,10,2
      IF(AT(I,J).EQ.0)GOTO 26
      AT(I,J+1)=ADJ(AT(I,J+1)-IA)
25    CONTINUE
26    AT(I,14)=ADJ(AT(I,14)-IA)
27    CONTINUE
      IFL(10)=IFL(11)=0
28    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SKIP(I)
      COMMON /C/ AT(150,20),DUM(250,4),ENG(20,3),ICHAR(200),KPER(20)
C*****************************************************************
C     ROUTINE TO MOVE IN 'AT' TABLE FOR AN ADDITIONAL PAIR OF LOCATIONS
C*****************************************************************
      DO 1 J=11,6,-1
1     AT(I,J)=AT(I,J-2)
      RETURN
      END
```

```
      FUNCTION ADJ  (X)
*****************************************************************
*     FUNCTION TO ADJUST  ANGLES BETWEEN 0 AND 360 DEGREES
*****************************************************************
      ADJ=X
1     IF(ADJ.GE.360.)ADJ=ADJ-360.;GOTO     1
2     IF(ADJ.LT.0.)ADJ=ADJ+360.;GOTO     2
      RETURN
      END
```

```
      SUBROUTINE RGSAN
      COMMON /B/ IBND, IFOR(80),ITA(40),IFL(20)
      COMMON /C/ AT(150,20),DRW(250,4),KRG(20,3),ICHAR(200),KPER(20)
      COMMON /D/ ISAC(40),IST,KLINK(20,2),KBRG(20)
      DIMENSION NX(20)
*************************************************************
*     ANALYSE AND ENTER IN TABLE 'AT' THE REST OF NOTATION
*     IN RING NOTATION MEANS OXYGEN ETC IN RING
*************************************************************
      LL=IFL(2);II=IFL(6)+1
1     LL=LL+1;IF(LL.GT.IFL(17))GOTO 16
      IF(IFOR(LL).LT.10)GOTO 18
      IF(IFOR(LL).GT.26)GOTO 2
      GOTO(3,16,15,18,18,5,18,18,18,18,18,6,18,16,7,18,8)IFOR(LL)-9
2     GOTO(4,10,4,18,18,10,16,11,13,131,14,5,18)IFOR(LL)-26
*************************************************************
*     ANALYSIS OF SPACE CHARACTER
*************************************************************
3     LL=LL+1;IF(LL.GT.IFL(17))GOTO 16
      IF(IFOR(LL).LT.14)GOTO 18
      II=IFL(6)+IFOR(LL)-13
      IF(IFOR(LL+1).EQ.11)LL=LL+1;II=II+23
      GOTO 1
*************************************************************
*     BORON AND PHOSPHOR AND NITROGEN ATOM
*************************************************************
4     CALL ICH(1,II,ITA(IFOR(LL)+1))
5     II=II+1
      GOTO 1
*************************************************************
*     HYDROGEN ATOM
*************************************************************
6     AT(II,12)=AT(II,12)-1
```

```
      GOTO 5
*****************************************************************
*     NITROGEN CHARGED ATOM
*****************************************************************
7     NX(1)=78;NX(2)=43
      CALL ICH(2,II,NX)
      GOTO 5
*****************************************************************
*     IMINO ATOM  M
*****************************************************************
8     AT(II,12)=0
      CALL ICH(1,II,78)
      IF(AT(II,4).NE.0)CALL SKIP(II)
      IFL(8)=IFL(8)+1
      CALL ICH(1,IFL(8),72)
9     AT(II,4)=IFL(8)
      AT(II,5)=AT(II,14)
      AT(IFL(8),14)=AT(II,14)
      AT(II,13)=AT(II,13)+1
      GOTO 5
*****************************************************************
*     OXYGEN AND SULFUR ATOM
*****************************************************************
10    AT(II,12)=0
      GOTO 4
*****************************************************************
*     UNSATURATION TAKE CARE OF U- LOC
*****************************************************************
11    AT(II,12)=AT(II,12)+1
      IF(IFOR(LL+1).EQ.12.AND.IFOR(LL+2).EQ.10)GOTO 12
      AT(II+1,12)=AT(II+1,12)+1
      GOTO 1
12    IX=IFOR(LL+3)+IFL(6)-13
```

```
      AT(IX,12)=AT(IX,12)+1
      LL=LL+3;GOTO 5
*************************************************************
*     CARBONYL V
*************************************************************
13    IF(AT(II,4).NE.0)CALL SKIP(II)
      AT(II,12)=1
      IFL(8)=IFL(8)+1
      CALL ICH(1,IFL(8),79)
      AT(IFL(8),12)=1
      GOTO 9
*************************************************************
*     W FOR DIOXYGEN
*************************************************************
131   II=II-1
      IF(AT(II,4).NE.0)CALL SKIP(II)
      AT(II,12)=1
      IFL(18)=IFL(18)+1
      NX(1)=79;NX(2)=2
      CALL ICH(2,IEL(8),NX)
      AT(IFL(8),12)=1
      GOTO 9
*************************************************************
*     SPIRO DETECTION
*************************************************************
14    IFL(9)=IFL(9)+1
      AT(II,18)=1
      GOTO 5
*************************************************************
*     METAL ELEMENT
*************************************************************
15    IF(IFOR(LL+1).LT.14.OR.IFOR(LL+2).LT.14.OR.IFOR(LL+3).NE.12)      W
     1GOTO 18
```

```
      NX(1)=ITA(IFOR(LL+1)+1);NX(2)=ITA(IFOR(LL+2)+1)
      LL=LL+3
      CALL ICH(2,II,NX)
151   IF(IFOR(LL+1).EQ.21)LL=LL+1;GOTO 151
      GOTO 5
*****************************************************************
*     RESTORE FLAG
*****************************************************************
16    IFL(2)=IFL(5)
X     PRINT 100
X     DO 17 I=1,IFL(8)
X     PRINT 101,I,AT(I,1),(AT(I,J),J=4,19)
X17   CONTINUE
X     PRINT 102,(ICHAR(I),I=1,IFL(15))
      RETURN
18    PRINT 103
      IFL(20)=1
      RETURN
X100  FORMAT(' TABLE AT END OF RGSAN')
X101  FORMAT(1X,I2,I3,4(I3,F5.0),2I3,F5.0,5I3)
X102  FORMAT(30I3)
103   FORMAT('ERROR DETECTED BY RGSAN')
      END
```

```
      SUBROUTINE RGPS
      COMMON /B/ IBND, IFOR(80),ITA(40),IFL(20)
      COMMON /C/ AT(150,20),DRW(250,4),KRG(20,3),ICHAR(200),KPER(20)
*****************************************************************
*     ROUTINE TO COMPOSE A DRAWING TABLE FROM THE CONNECTING TABLE
*     THE DRAWING TABLE IS A N BY 4 MATRIX WHERE COLUMNS 1 & 2 ARE
*     THE ABSOLUTE COORDINATES WHERE THE BEAM HAVE TO MOVE (WITHOUT DRAW)
*     AND COLUMNS 3 & 4 THE ABSOLUTE COORDINATES OF THE DRAW.
*****************************************************************
      PI=57.2957795
      X=500.;Y=365.;Z=60.
      AT(1,2)=X;AT(1,3)=Y
      K=0
      REPEAT 3, FOR I=(1,IFL(8))
      IF(AT(I,2).EQ.0)GOTO 3
      DO 2 J=4,10,2
      IF(AT(I,J).EQ.0)GOTO 3
      K=K+1
      IX=AT(I,J)
      Z=60.
      IF(AT(I,17).EQ.1.AND.AT(IX,17).EQ.1)Z=50.
      AX=SIN(AT(I,J+1)/PI)
      AY=COS(AT(I,J+1)/PI)
      IF(AT(IX,2).NE.0.AND.AT(IX,3).NE.0)GOTO 1
      AT(IX,2)=AT(I,2)+Z*AX
      AT(IX,3)=AT(I,3)+Z*AY
      IF(IX.LT.I)I=IX-1;GOTO 3
1     CALL BOX(AT(I,2),AT(I,3),AT(IX,2),AT(IX,3),DRW(K,1),DRW(K,2),DRW(K
     1,3),DRW(K,4),8,AT(I,15)*5,AT(IX,15)*5)
      IF(AT(I,12).EQ.0.OR.AT(IX,12).EQ.0)GOTO 2
      K=K+1
      AA=AT(I,2)+5.*SIN((AT(I,J+1)+90.)/PI)
      AB=AT(I,3)+5.*COS((AT(I,J+1)+90.)/PI)
```

```
      DRW(K,1)=AA+15.*AX
      DRW(K,2)=AB+15.*AY
      DRW(K,3)=AA+(Z-15.)*AX
      DRW(K,4)=AB+(Z-15.)*AY
      AT(I,12)=AT(I,12)-1
      AT(IX,12)=AT(IX,12)-1
2     CONTINUE
3     CONTINUE
      IFL(19)=K
X     PRINT 100
X     DO 4  I=1,IFL(8)
X     PRINT 101,I,AT(I,2),AT(I,3)
X4    CONTINUE
X     PRINT 102
X     DO 5  I=1,IFL(19)
X     PRINT 103,I,(DRW(I.J),J=1,4)
X5    CONTINUE
X100  FORMAT(' AT END OF RGPS TABLE AT')
X101  FORMAT(1X,I3,2F5.0)
X102  FORMAT(' NOW TABLE DRW')
X103  FORMAT(1X,I3,4F6.0)
      RETURN
      END
```

```
      SUBROUTINE DRAG
      COMMON /B/ IBND, IFOR(80),ITA(40),IFL(20)
      COMMON /C/ AT(150,20),DRW(250,4),KRG(20,3),ICHAR(200),KPER(20)
*****************************************************************
*     ROUTINE TO DRAW THE STRUCTURE ON A TEKTRONIX DISPLAY TERMINAL
*     THIS HAVE TO BE MODIFIED FOR AN OTHER DISPLAY TERMINAL
*****************************************************************
      DO 1 I=1,IFL(19)
      CALL MOVEA(DRW(I,1),DRW(I,2))
      CALL DRAWA(DRW(I,3),DRW(I,4))
1     CONTINUE
      DO 3 I=1,IFL(8)
      IF(AT(I,15).EQ.0)GOTO 3
      CALL MOVEA(AT(I,2)-AT(I,15)*5,AT(I,3)-4)
      DO 2 J=1,AT(I,15)
      IC=ICHAR(AT(I,16)-1+J)
      CALL ANCHO(IC)
2     CONTINUE
3     CONTINUE
      CALL HOME
      CALL ANMODE
      RETURN
      END
```

```
      SUBROUTINE CHAIN
      COMMON /B/ IBND, IFOR(80),ITA(40),IFL(20)
      COMMON /C/ AT(150,20),DRW(250,4),KRG(20,3),ICHAR(200),KPER(20)
      COMMON /D/ ISAC(40),IST,KLINK(20,2),KBRG(20)
      DIMENSION NX(20)
C*************************************************************
C     ANALYSIS OF THE CHAIN, CHAIN OF ELEMENTS OR CHAIN OF RINGS
C*************************************************************
      ICX=IX=0;MZ=IFL(8);ISPR=0
      LL=IFL(2);II=IFL(6)+1
      IF(LL.EQ.0)AT(1,14)=60
1     LL=LL+1;IF(LL.GT.IFL(3))GOTO 270
      JH=0
      IF(IFOR(LL).GT.24)GOTO 2
      IF(IFOR(LL).LT.10)GO TO 10
      GOTO(3,30,40,50,60,70,80,90,100,110,120,130,110,60,140)IFOR(LL)-9
2     GOTO (90,150,70,160,70,170,180,160,90,190,200,210,220,230,240)
     1IFOR(LL)-24
C*************************************************************
C     ANALYSIS OF SPACE CHARACTER
C*************************************************************
3     LL=LL+1;IF(LL.GT.IFL(3))GOTO 270
C*************************************************************
C     IF FOR MULTIPLIER DETECTION
C*************************************************************
      IF(IFOR(LL).LT.14)GOTO 250
C*************************************************************
C     COMPUTE THE LOCANT PATH
C*************************************************************
      II=ILOC(LL)+IFL(6)
C*************************************************************
C     DETECT A '-' FOR NEW RING LOCATION
C*************************************************************
```

```
      IF(II.GT.1000)II=II-(II/100)*100;GOTO 40
*****************************************************************
*     CASE OF METHYL REDUCTION
*****************************************************************
      IBND=1
      IF(IFOR(LL+1).EQ.10)CALL REDUCE(3)
      GOTO 1
*****************************************************************
*     ANALYSE OF CARBON ATOM ANY COMBINAISON OF NUMBER AND U
*****************************************************************
10    IF(IFOR(LL+1).GT.9)GOTO 11
*****************************************************************
*     SEE IF MORE THAN 9 CARBONS ON CHAIN
*****************************************************************
      LL=LL+1;IFOR(LL)=IFOR(LL)+IFOR(LL-1)*10
11    IF(IBND.GT.0)GOTO 16
*****************************************************************
*     1 2 AS FIRST ELEMENT
*****************************************************************
      MZ=MZ+1
      IBND=1
12    JH=JH+1;NX(JH)=67
      JH=JH+1;NX(JH)=72
*****************************************************************
*     BRANCH FOR 1U CASE
*****************************************************************
      IF(IFOR(LL+1).EQ.34.AND.IFOR(LL).EQ.1)GOTO 15
      JH=JH+1;NX(JH)=51
      IF(IFOR(LL).EQ.1)GOTO 27
      IFOR(LL)=IFOR(LL)-1
13    IF(IFOR(LL+1).NE.34)GO TO 14
*****************************************************************
*     CASE OF 2U OR 3U OR ...
```

```
*****************************************************************
      IX=-1;IFOR(LL)=IFOR(LL)-1
      IF(IFOR(LL).EQ.0)GOTO 22
      GOTO 18
14    IF(IFOR(LL+1).NE.11.AND.IFOR(LL+1).NE.10)GOTO 18
      IX=1;IFOR(LL)=IFOR(LL)-1
      IF(IFOR(LL).EQ.0)GOTO 25
      GOTO 18
15    IF(IFOR(LL+2).EQ.34)GOTO 27
      JH=JH+1;NX(JH)=50
      GOTO 27
16    ICX=1;CALL ENTRX(1)
      IF(IBND.GT.1)GOTO 20
      IF(IFOR(LL+1).NE.34)GOTO 17
      IF(IFOR(LL).EQ.1)GOTO 22
      GOTO 13
17    IF(IFOR(LL+1).EQ.11.OR.IFOR(LL+1).EQ.10)GOTO 13
18    IF(IFOR(LL).GT.1)JH=JH+1;NX(JH)=40
*****************************************************************
*     ALIPHATIC CHAIN
*****************************************************************
      JH=JH+1;NX(JH)=67
      JH=JH+1;NX(JH)=72
      JH=JH+1;NX(JH)=50
      IF(IFOR(LL).EQ.1)GOTO 19
      JH=JH+1;NX(JH)=41
      IF(IFOR(LL).LT.10)JH=JH+1;NX(JH)=ITA(IFOR(LL)+1);GOTO 19
      JH=JH+1;NX(JH)=ITA(IFOR(LL)/10+1)
      JH=JH+1;NX(JH)=ITA(IFOR(LL)-(IFOR(LL)/10)*10+1)
19    IF(IX)22,27,25
20    IF(IFOR(LL).EQ.1)GOTO 23
      IFOR(LL)=IFOR(LL)-1
      JH=JH+1;NX(JH)=67
```

```
      IF(IBND.GT.2)IBND=1;GOTO 13
      JH=JH+1;NX(JH)=72
      IBND=1;GOTO 13
21    JH=JH+1;NX(JH)=67
      IF(IBND.GT.2)IBND=1;GOTO 27
      JH=JH+1;NX(JH)=72
      IBND=1;GOTO 27
22    IX=0
      JH=JH+1;NX(JH)=67
      IF(IFOR(LL+2).EQ.34)GOTO 27
      JH=JH+1;NX(JH)=72
      GOTO 27
23    IF(IFOR(LL+1).EQ.11.OR.IFOR(LL+1).EQ.10)GOTO 26
      IF(IFOR(LL+1).NE.34)GOTO 21
      IF(IFOR(LL+2).EQ.34.OR.IBND.GT.1)GOTO 260
      JH=JH+1;NX(JH)=67
      GOTO 27
24    IF(IFOR(LL).EQ.1)GOTO 12
      GOTO 14
25    IFOR(LL)=1;IX=0
      GOTO 12
26    JH=JH+1;NX(JH)=67
      JH=JH+1;NX(JH)=72
      IF(IBND.GT.1)IBND=1;GOTO 27
      JH=JH+1;NX(JH)=50
      GOTO 27
27    II=MZ
28    CALL ICH(JH,MZ,NX)
      GOTO 1
*****************************************************************
*     ANALYSIS OF &
*****************************************************************
30    IF(IST.LT.0)IFL(6)=II=0;GOTO 1
```

```
      IF(ISAC(IST).LT.0)GOTO 32
      IST=IST-1;IF(IST.EQ.0)IFL(6)=II=0;GOTO 1
      DO 31 I=IST,1,-1
      IF(ISAC(I).LT.0)GOTO   31
      II=IFL(6)=ABS(ISAC(IST))
      GOTO 1
31    CONTINUE
      IFL(6)=II=0
32    CALL REDUCE(2)
      GOTO 1
*****************************************************************
*     ANALYSIS OF -
*****************************************************************
40    IF(IFOR(LL+1).NE.10.AND.IFOR(LL+1).NE.11)GOTO 41
      IFL(11)=II
      IST=IST+1;ISAC(IST)=II
      IF(IFOR(LL+1).EQ.11)LL=LL+1;ISPR=1
      GOTO 1
41    CALL ENTRX(0)
      LL=LL+1;IF(LL.GT.IFL(3))GOTO 270
      JH=JH+1;NX(JH)=ITA(IFOR(LL)+1)
      LL=LL+1;IF(LL.GT.IFL(3))GOTO 270
      IF(IFOR(LL).EQ.12)GOTO 27
      JH=JH+1;NX(JH)=ITA(IFOR(LL)+1)
      LL=LL+1;IF(LL.GT.IFL(3))GOTO 270
      IF(IFOR(LL).EQ.12)GOTO 27
      GOTO 260
*****************************************************************
*     ANALYSIS OF /
*****************************************************************
50    IF(IFL(16).EQ.0)IFL(16)=1;IFL(17)=1;GOTO 1
      IF(IFOR(LL-1).EQ.13)GOTO 1
51    LL=LL+1;IF(LL.GT.IFL(3))GOTO 270
```

```
      IF(IFOR(LL).NE.10)GOTO 260
      LL=LL+1;IF(LL.GT.IFL(3))GOTO 270
      IF(IFOR(LL).GT.9)GOTO 260
      JH=JH+1;NX(JH)=ITA(IFOR(LL)+1)
      IF(IFOR(LL+1).EQ.10)GOTO 51
      CALL ICH(JH,MZ,NX)
      GOTO 1
*****************************************************************
*     ALKYL
*****************************************************************
60    CALL ENTRX(0)
      JH=JH+1;NX(JH)=88
61    IF(LL.EQ.1)IBND=1;GOTO 27
62    CALL ICH(JH,MZ,NX)
      ICX=0;IBND=1
      IF(IST.EQ.0)GOTO 1
      IF(ISAC(IST).GE.0)GOTO 1
      II=ABS(ISAC(IST))
      IST=IST-1
      GOTO 1
*****************************************************************
*     ANALYSIS OF BORON AND N
*****************************************************************
70    CALL ENTRX(2)
      IF(IFOR(LL).EQ.29)IFL(12)=MZ
      JH=JH+1;NX(JH)=ITA(IFOR(LL)+1)
      IF(IBND.EQ.2)IBND=1;GOTO 27
      JK=1;GOTO 141
*****************************************************************
*     ANALYSE OF C
*****************************************************************
80    CALL ENTRX(1)
      IF(IFOR(LL-1).EQ.27)IST=IST-1
```

```
      JH=JH+1;NX(JH)=67
      GOTO 27
*****************************************************************
#     ANALYSE OF D
*****************************************************************
90    IFL(10)=II-IFL(6);LL=LL-1
      IF(IST.EQ.0)GOTO 91
      II=ABS(ISAC(IST))
      IST=IST-1
91    IF(ISPR.EQ.1)AT(IFL(11),20)=IFL(10);GOTO 93
      MZ=MZ+IFL(10)-1
      CALL ENTRX(0)
      MZ=MZ-IFL(10)
93    IST=IST+1;ISAC(IST)=MZ
92    IFL(8)=MZ;IFL(2)=LL
      RETURN
*****************************************************************
#     ANALYSIS OF E
*****************************************************************
100   CALL ENTRX(0)
      JH=JH+1;NX(JH)=66
      JH=JH+1;NX(JH)=82
      GO TO 61
*****************************************************************
#     ANALYSIS OF F
*****************************************************************
110   CALL ENTRX(0)
      JH=JH+1;NX(JH)=ITA(IFOR(LL)+1)
      GO TO 61
*****************************************************************
#     ANALYSIS OF G
*****************************************************************
120   CALL ENTRX(0)
```

```
      JH=JH+1;NX(JH)=67
      JH=JH+1;NX(JH)=76
      GO TO 61
*****************************************************************
*     ANALYSIS OF H
*****************************************************************
130   JH=JH+1;NX(JH)=ITA(IFOR(LL)+1)
      GOTO 62
*****************************************************************
*     ANALYSIS OF K
*****************************************************************
140   CALL ENTRX(3)
      JH=JH+1;NX(JH)=78
      JH=JH+1;NX(JH)=43
      IF(IBND.GT.2)IBND=1;GOTO 27
      IF(IBND.EQ.2)JK=1;GOTO 141
      JK=2
141   DO 142 I=1,JK
      IST=IST+1
142   ISAC(IST)=-MZ
      IBND=1
      GOTO 27
*****************************************************************
*     ANALYSIS OF M
*****************************************************************
150   CALL ENTRX(1)
      JH=JH+1;NX(JH)=78
      JH=JH+1;NX(JH)=72
      GOTO 27
*****************************************************************
*     ANALYSIS OF O
*****************************************************************
160   CALL ENTRX(1)
```

```
      JH=JH+1;NX(JH)=ITA(IFOR(LL)+1)
      GOTO 27
*****************************************************************
*     ANALYSIS OF Q
*****************************************************************
170   CALL ENTRX(0)
      JH=JH+1;NX(JH)=79
      JH=JH+1;NX(JH)=72
      GO TO 61
*****************************************************************
*     ANALYSIS OF R
*****************************************************************
180   CALL ENTRX(0)
      XX=AT(MZ,14)-120
      IST=IST+1;ISAC(IST)=MZ-1
      DO 181 I=1,6
      AT(MZ,4)=MZ+1
      AT(MZ,5)=ADJ(XX+60*I)
      AT(MZ,14)=ADJ(AT(MZ,5)-120)
      AT(MZ,12)=1;MZ=MZ+1
181   CONTINUE
      MZ=MZ-1
      AT(MZ,4)=ISAC(IST)+1
      IFL(6)=ISAC(IST)
      GOTO 1
*****************************************************************
*     ANALYSE OF U
*****************************************************************
190   AT(II,12)=AT(II,12)+1
      AT(MZ+1,12)=AT(MZ+1,12)+1
      IBND=IBND+1
      GOTO 1
*****************************************************************
```

```
*     ANALYSE OF V
*****************************************************************
200   CALL ENTRX(1)
      JH=JH+1;NX(JH)=67
      JH=JH+1;NX(JH)=79
      GOTO 27
*****************************************************************
*     ANALYSE OF W
*****************************************************************
210   CALL ENTRX(0)
      JH=JH+1;NX(JH)=79
      JH=JH+1;NX(JH)=50
      IF(LL.EQ.1)IBND=2;GOTO 27
      IF(IST.EQ.0)GOTO 62
      IF(ABS(ISAC(IST)).EQ.II)IST=IST-1
      GOTO 62
*****************************************************************
*     ANALYSE OF X
*****************************************************************
220   CALL ENTRX(3)
      JH=JH+1;NX(JH)=67
      IF(IBND.GT.2)IBND=1;GOTO 27
      IF(IBND.EQ.2)JK=1;GOTO 141
      JK=2
      GOTO 141
*****************************************************************
*     ANALYSE OF Y
*****************************************************************
230   CALL ENTRX(2)
      JH=JH+1;NX(JH)=67
      JK=1;GOTO 141
*****************************************************************
*     ANALYSE OF Z
```

```
*****************************************************************
240   CALL ENTRX(0)
      JH=JH+1;NX(JH)=78
      JH=JH+1;NX(JH)=72
      JH=JH+1;NX(JH)=50
      GOTO 6
*****************************************************************
*     ANALYSIS OF CONTACTION
*****************************************************************
250   IF(IFOR(LL).NE.11)GOTO 260
*****************************************************************
*     SALT
*****************************************************************
      CALL ENTRX(0)
      JH=JH+1;NX(JH)=40
      JH=JH+1;NX(JH)=45
      JH=JH+1;NX(JH)=43
      JH=JH+1;NX(JH)=41
      GOTO 27
260   PRINT 261
261   FORMAT('ERROR IN CHAIN')
      IFL(20)=1
      RETURN
270   CALL REDUCE(1)
      IFL(8)=MZ;IFL(2)=LL
X     PRINT 1000
X1000 FORMAT(' AT END OF CHAIN TABLE AT')
X     DO 1005 I=1,IFL(8)
X     PRINT 1001,I,(AT(I,J),J=4,20)
X1001 FORMAT(1X,I2,4(I3,I4),2I2,I4,8I3)
X1005 CONTINUE
      RETURN
*****************************************************************
```

```
      SUBROUTINE REDUCE(N)
      DIMENSION NN(3)
      DATA NN/67,72,51/
*************************************************************
*     ROUTINE TO ANALYSE THE STACK AT THE END OF A CHAIN
*     TAKE CARE OF METHYL REDUCTION
*************************************************************
      GOTO(1,3,2)N
1     IF(IST.LT.1)RETURN
      IF(ISAC(IST).GE.0)RETURN
      IF(ICX.GT.0)ICX=0;IST=IST-1;GOTO 1
      II=ABS(ISAC(IST))
      CALL ENTRX(0)
      CALL ICH(JH,MZ,NN)
      IST=IST-1
      GOTO 1
2     CALL ENTRX(0)
      JH=3
      CALL ICH(JH,MZ,NN)
      RETURN
3     IF(IFOR(LL-1).EQ.12.AND.IFL(12).NE.0)IFL(12)=0;GOTO 4
      IF(IFOR(LL-1).EQ.38.OR.IFOR(LL-1).EQ.37.OR.IFOR(LL-1).EQ.24.OR.
     1IFOR(LL-1).EQ.12)GOTO 5
4     II=ABS(ISAC(IST))
      IST=IST-1
      RETURN
5     CALL ENTRX(0)
      CALL ICH(JH,MZ,NN)
      GOTO 4
*************************************************************
      SUBROUTINE ENTRX(NL)
      DIMENSION AA(3,3)
      DATA AA/0,-60,0,-60,60,-90,60,0,90/
```

```
C*************************************************************
C     ROUTINE TO ENTER IN CONNECTING TABLE
C*************************************************************
      IF(AT(II,4).NE.0)CALL  SKIP(II)
      AT(II,4)=MZ=MZ+1
      IF(AT(MZ,17).EQ.0)AT(MZ,17)=1
      AT(MZ,19)=NL
      AT(II,13)=AT(II,13)+1
      IF(AT(II,19).EQ.0)AT(II,19)=1
      IF(AT(II,13).GT.3)AT(II,13)=2
      AT(II,5)=AT(II,14)+AA(AT(II,19),AT(II,13))
      AT(MZ,14)=AT(II,5)
      RETURN
      END
```

```
      SUBROUTINE ICH(N,II,NCH)
      DIMENSION NCH(1)
      COMMON /C/ AT(150,20),DRW(250,4),KRG(20,3),ICHAR(200),KPER(20)
      COMMON /B/ IBND, IFOR(80),ITA(40),IFL(20)
*****************************************************************
*     ROUTINE TO ENTER THE CHARACTERS TO BE DISPLAYED IN CHARACTER STACK
*     AND POINTER FOR THIS STACK IN CONNECTING TABLE
*****************************************************************
      AT(II,15)=N
      DO 1 I=1,N
1     ICHAR(IFL(15)+I)=NCH(I)
      AT(II,16)=IFL(15)+1
      IFL(15)=IFL(15)+N
      RETURN
      END
```

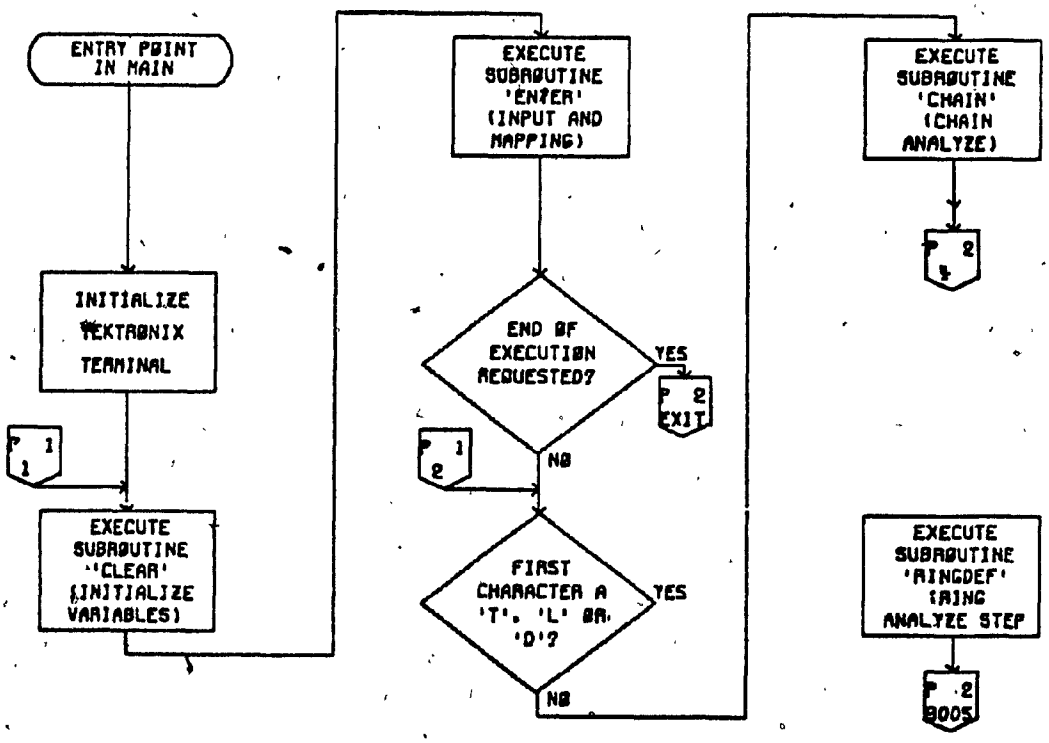
```
      SUBROUTINE BOX(X1,Y1,X2,Y2,X3,Y3,X4,Y4,IH,W1,W2)
*************************************************************
*     ROUTINE TO COMPUTE AN IMAGINARY BOX AROUND A STRING OF CHARACTERS AND
*     AND COMPUTE THE BOUNDARY COORDINATES FOR THE DRAWING TABLE
*************************************************************
      X=X2-X1
      Y=Y2-Y1
      IF(X.EQ.0.OR.Y.EQ.0)GOTO 2
      DH=IH*(Y/ABS(Y))
      DW=X/ABS(X)
      TAN=ABS(Y/X)
      Y3=Y1+DH
      X4=X3=ABS(DH/TAN)
      IF(W1.EQ.0)X3=X1;Y3=Y1;GOTO 1
      IF(X3.LT.W1)X3=X1+DW*X3;GOTO 1
      Y3=Y1+DH*TAN
      X3=X1+DW*W1
1     IF(W2.EQ.0)X4=X2;Y4=Y2;RETURN
      Y4=Y2-DH
      IF(X4.LT.W2)X4=X2-DW*X4;RETURN
      Y4=Y2-DH*TAN
      X4=X2-DW*W2
      RETURN
2     IF(Y.NE.0)GOTO 4
      Y3=Y4=Y1
      IF(W1.EQ.0)X3=X1;GOTO 3
      X3=X1+(X/ABS(X))*W1
3     IF(W2.EQ.0)X4=X2;RETURN
      X4=X2-(X/ABS(X))*W2
      RETURN
4     X3=X4=X1
      IF(W1.EQ.0)Y3=Y1;GOTO 5
      Y3=Y1+(Y/ABS(Y))*IH
```

```
5     IF(W2.EQ.0)Y4=Y2;RETURN
      Y4=Y2-(Y/ABS(Y))*IH
      RETURN
      END
```

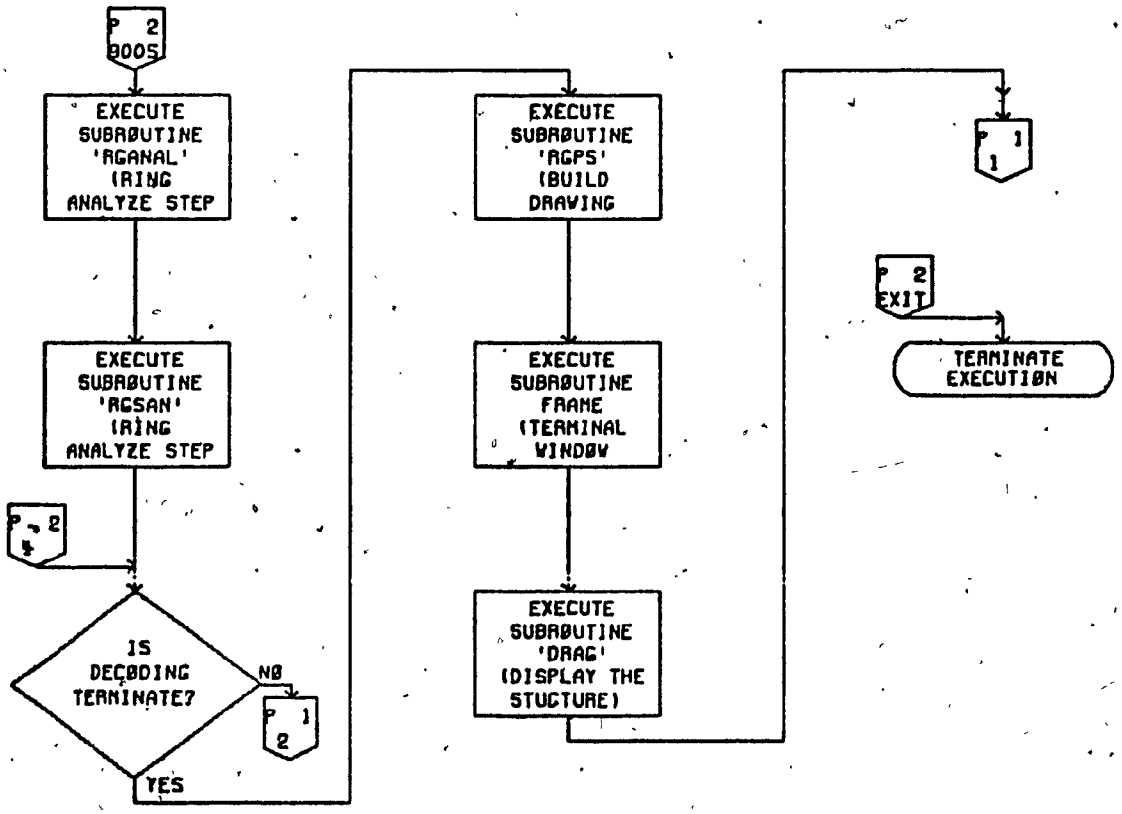
```
      SUBROUTINE FRAME
      COMMON /C/ AT(150,20),DRW(250,4),KRG(20,3),ICHAR(200),KPER(20)
      COMMON /B/ IBND, IFOR(80),ITA(40),IFL(20)
*****************************************************************
*     ROUTINE TO COMPUTE THE LIMIT OF THE PICTURE AND FIXE THE WINDOW
*****************************************************************
      XMAX=YMAX=0
      XMIN=YMIN=1E6
      DO 1 I=1,IFL(8)
      IF(AT(I,2).GT.XMAX)XMAX=AT(I,2)
      IF(AT(I,3).GT.YMAX)YMAX=AT(I,3)
      IF(AT(I,2).LT.XMIN)XMIN=AT(I,2)
      IF(AT(I,3).LT.YMIN)YMIN=AT(I,3)
1     CONTINUE
      XN=(XMAX-XMIN)/10
      IF(XN.LT.60.)XN=60.
      XM=XN*12
      XN=XMIN-XN
      YN=(YMAX-YMIN)/10
      IF(YN.LT.60.)YN=60.
      YM=YN*12
      YN=YMIN-YN
X     GOTO 2
      CALL VWINDO(XN,XM,YN,YM)
X2    PRINT 100,XMAX,XMIN,YMAX,YMIN
X100  FORMAT('FOR DISPLAY X MAX =',F5.0,' X MIN =',F5.0,' Y MAX =',
X    1F5.0,' Y MIN =',F5.0)
X     PRINT 101,XN,XM,YN,YM
X101  FORMAT('WINDOW IS ',F5.0,',',F5.0,',',F5.0,',',F5.0)
      RETURN
      END
```

APPENDIX B PROGRAM FLOWCHART

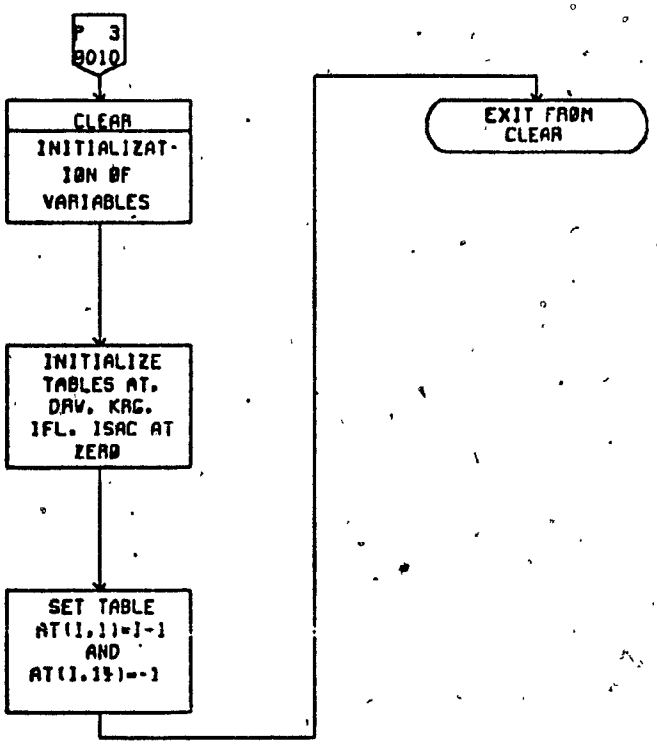
MAIN PRØGRAM



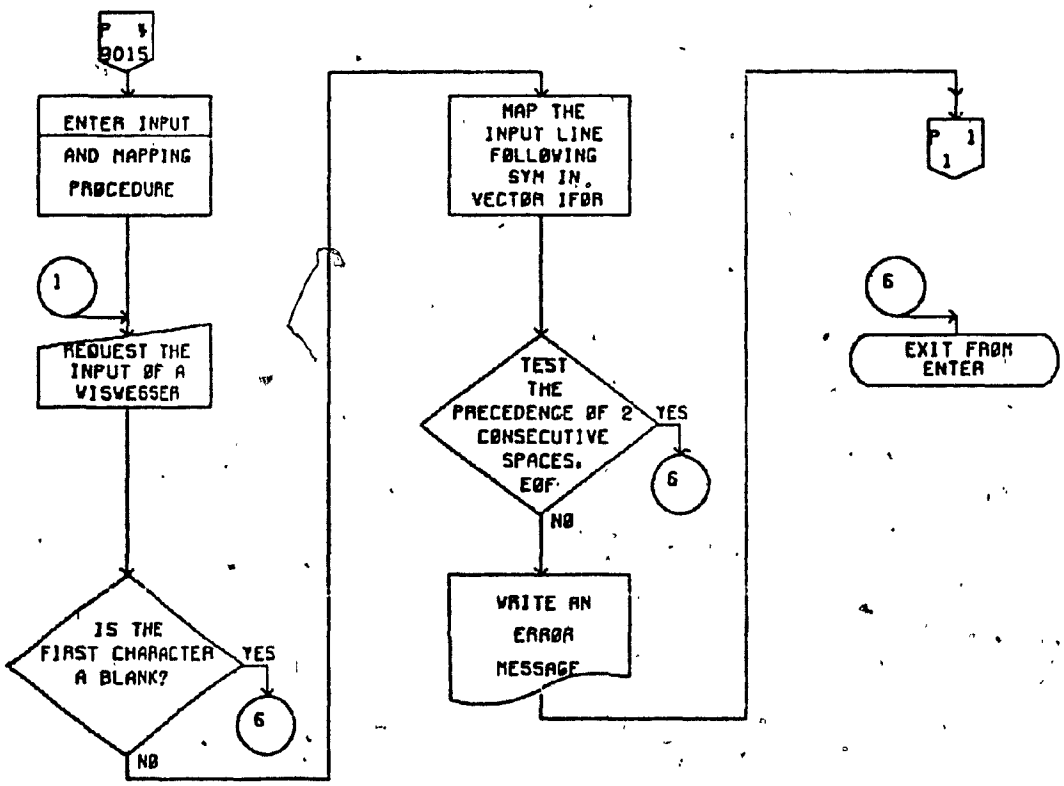
MAIN PRØGRAM



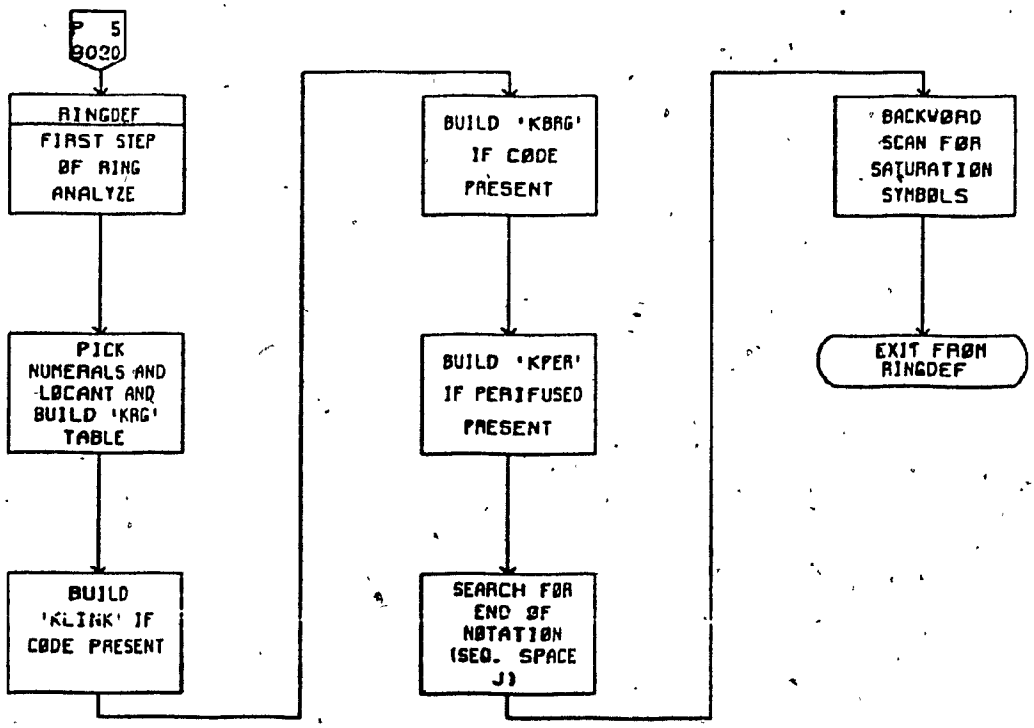
SUBRØUTINE CLEAR



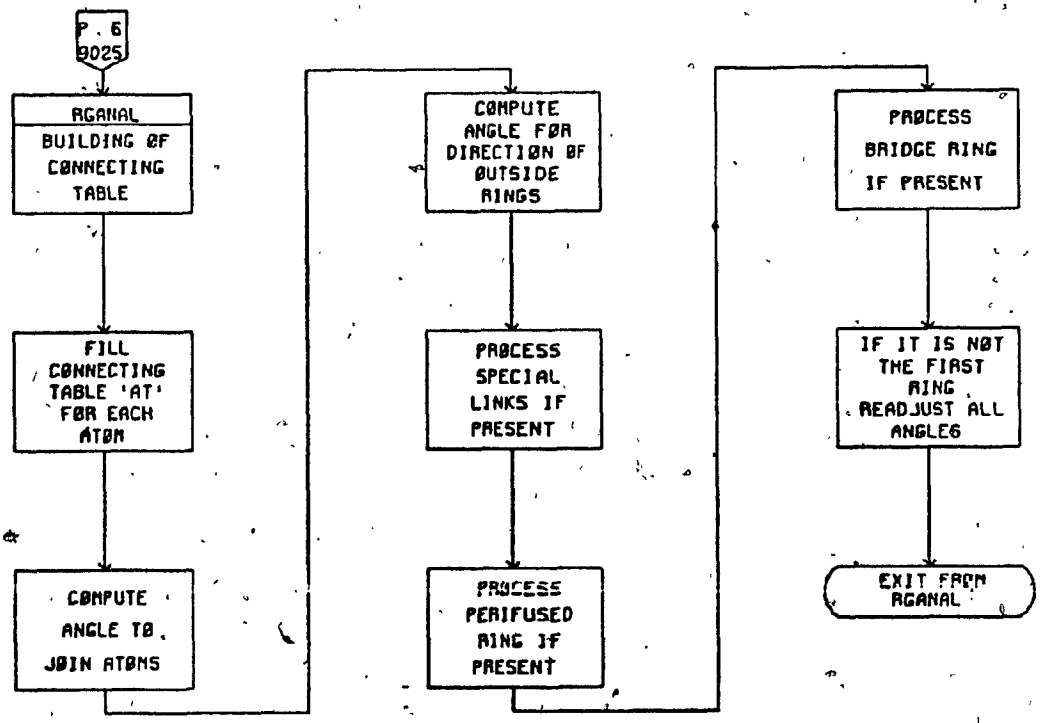
SUBRØUTINE ENTER



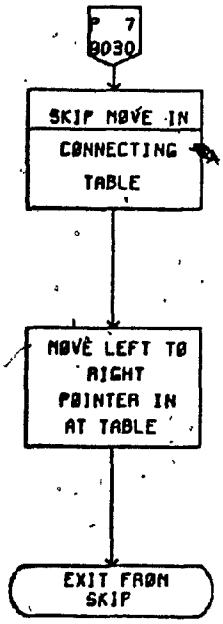
SUBRØUTINE RINGDEF



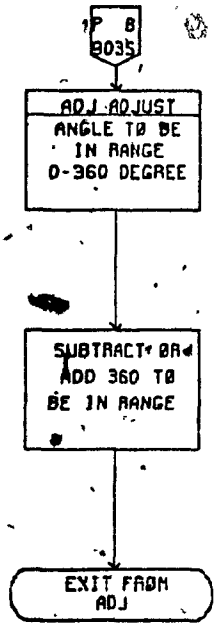
SUBRØUTINE RGANAL



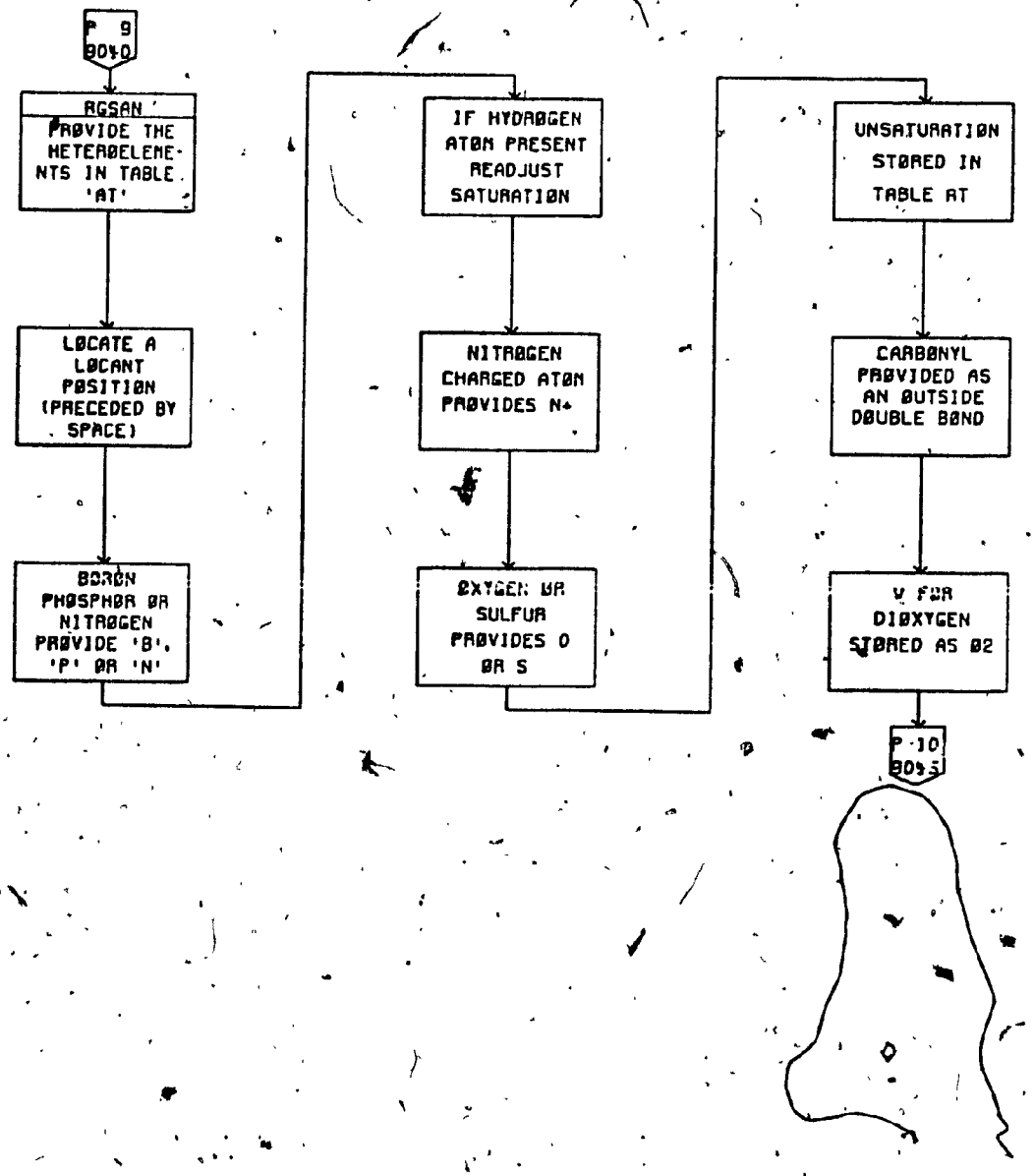
SUBRØUTINE SKIP



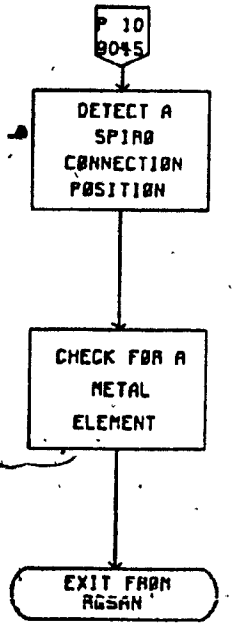
FUNCTIØN ADJ



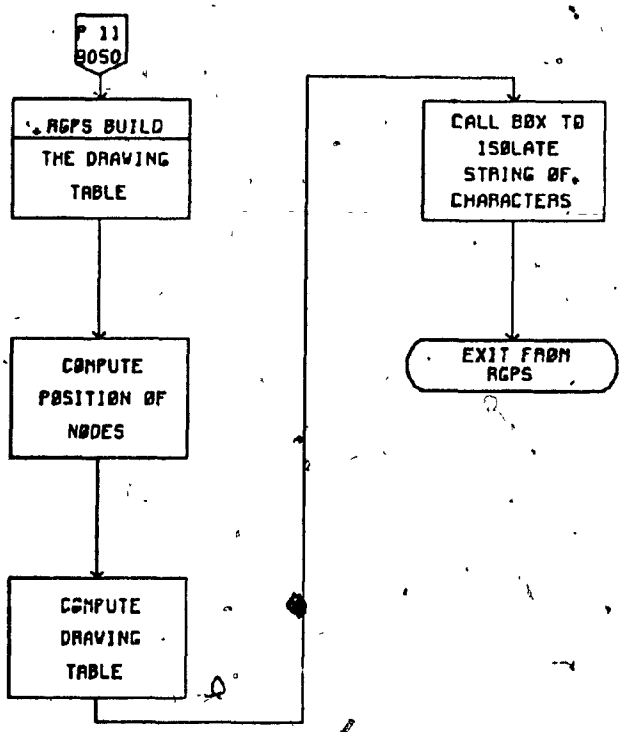
SUBRØUTINE RGSAN



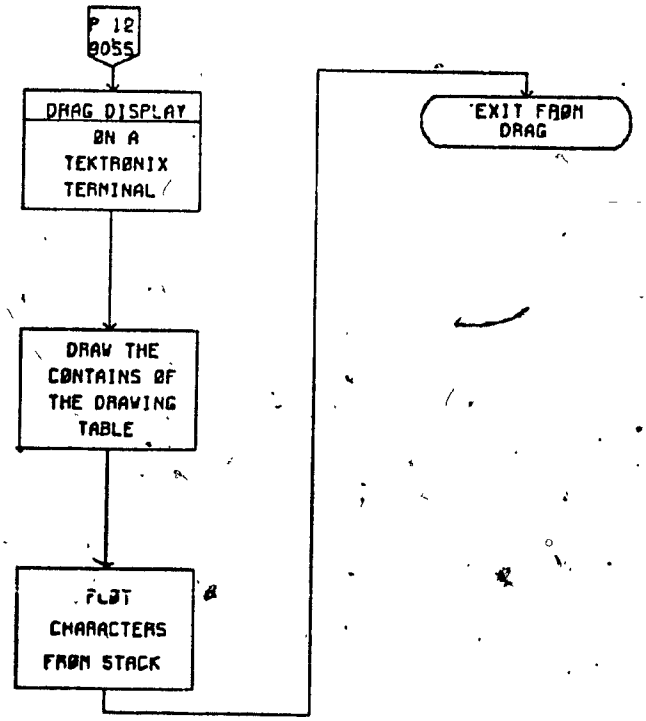
SUBRØUTINE RGSAN



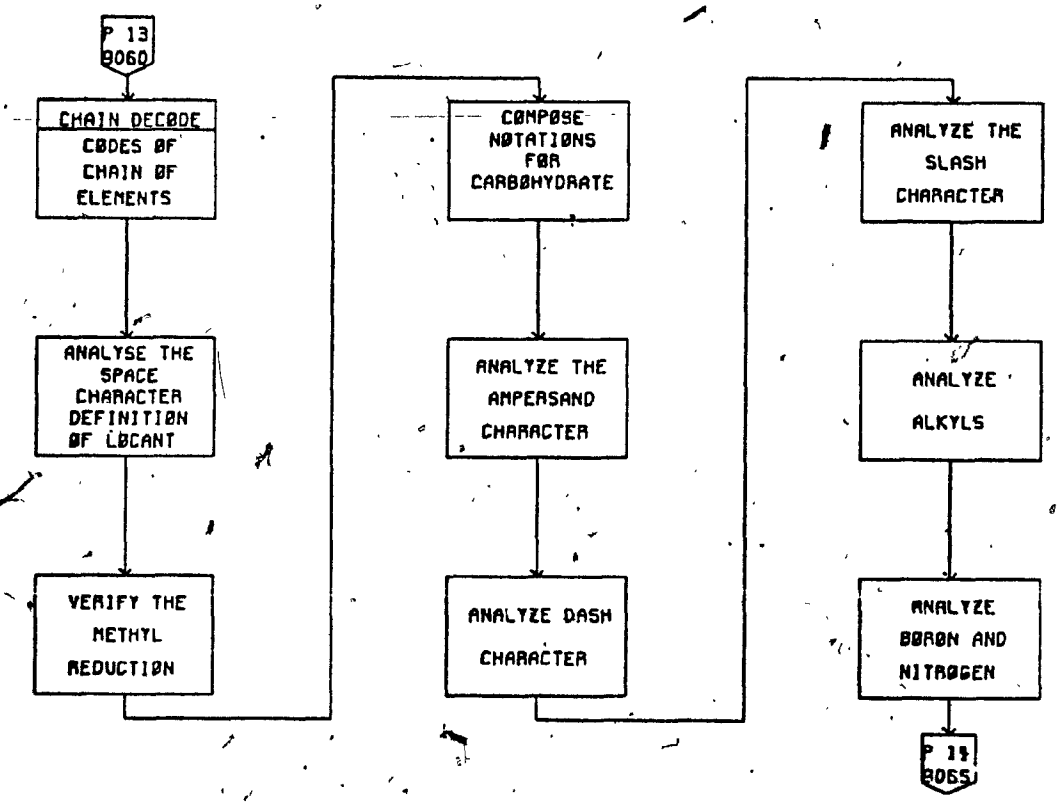
SUBRØUTINE RGPS



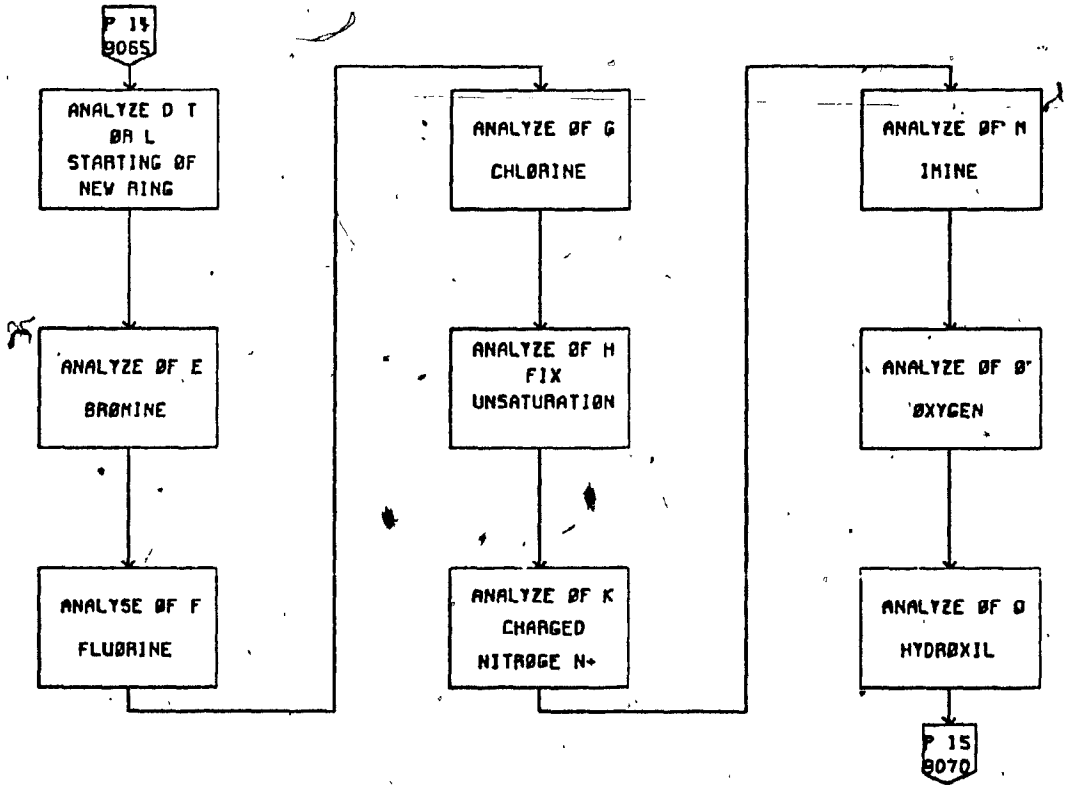
SUBRØUTINE DRAG.



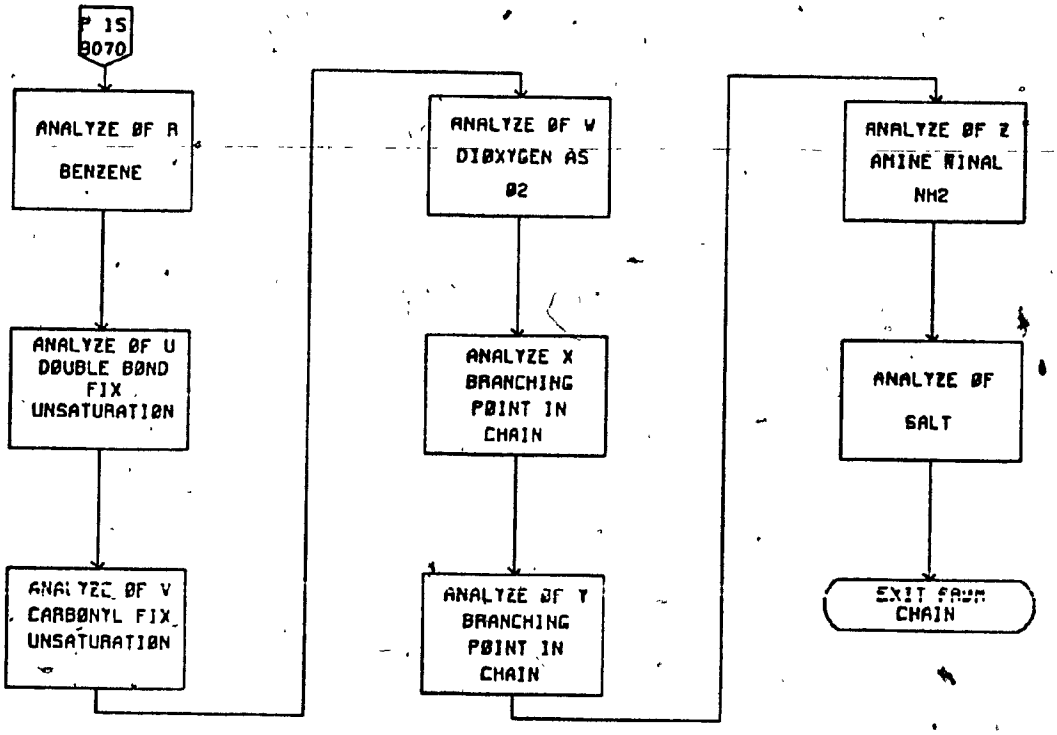
SUBRØUTINE CHAIN



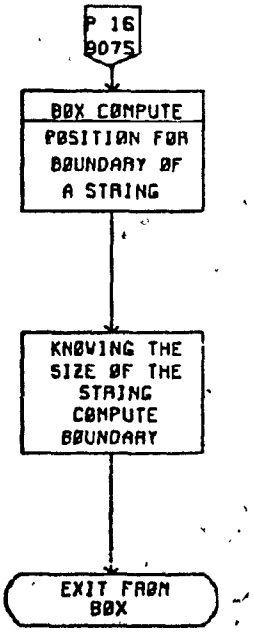
SUBROUTINE CHAIN



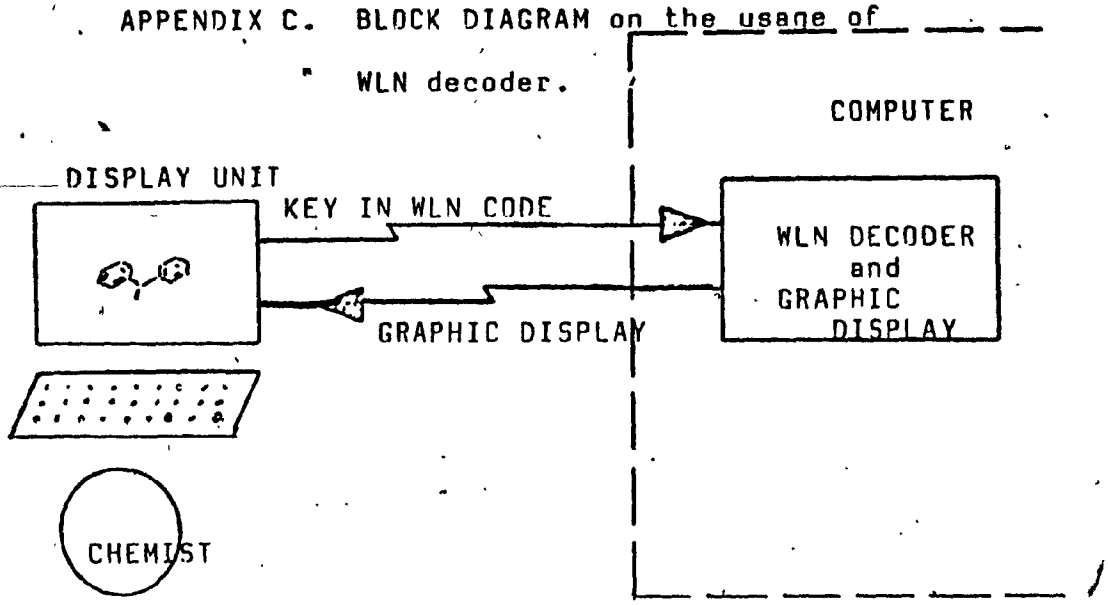
SUBRØUTINE CHAIN



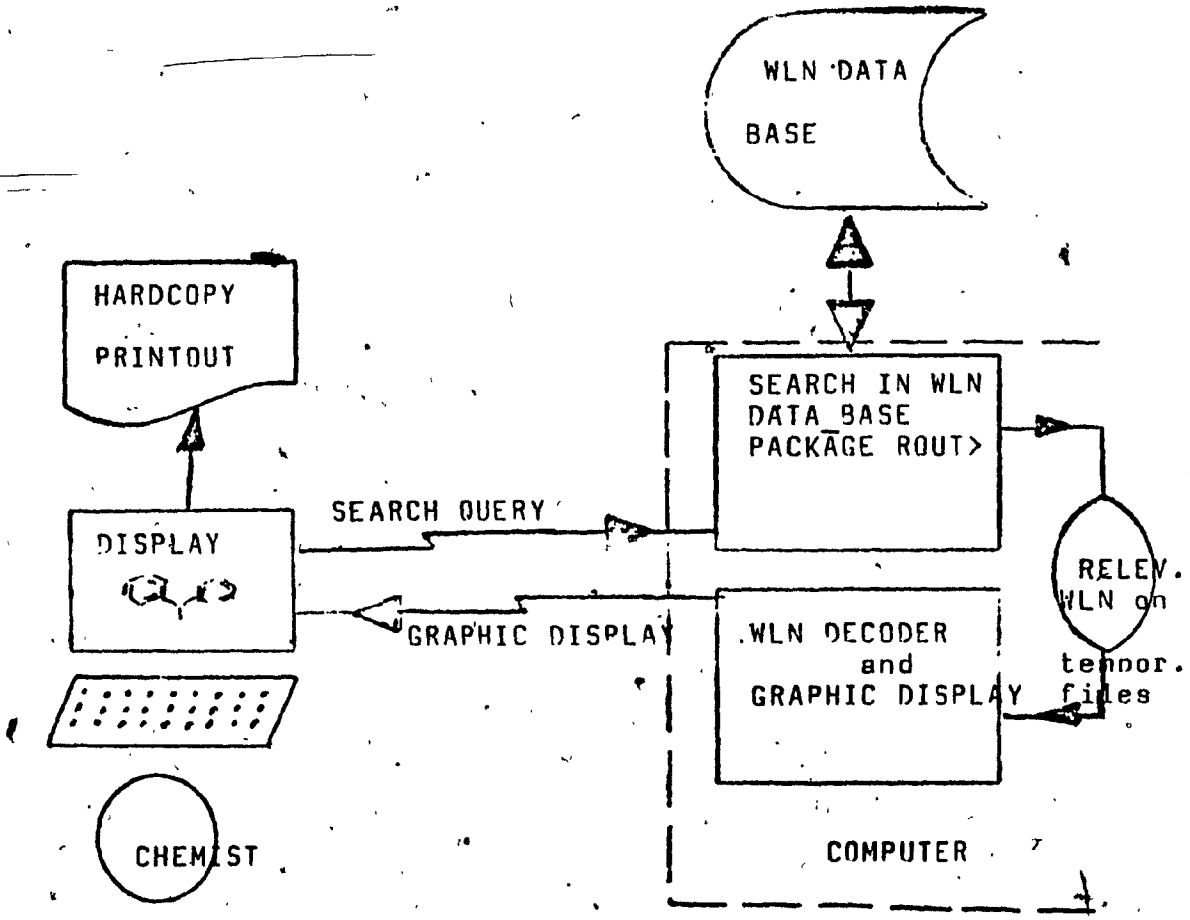
SUBRØUTINE BØX



APPENDIX C. BLOCK DIAGRAM on the usage of WLN decoder.



1. Verification of a manual coding of a WLN.



2. Graphic output of search in WLN files.